

**FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT**

**UNIVERSITI MALAYA**

**SESSI 2003/2004**

**Perpustakaan SKTM**

**LATIHAN ILMIAH II  
WXES 3182**

**MOHADI BIN MOHAMED  
WEK 98336**



**SISTEM ANALISIS PEMARKAHAN**

**PENYELIA : EN. ZAIDI**

**MODERATOR : NOORZAILLY**

**SYARAT BAGI PENGANUGERAHAN  
IJAZAH SARJANA MUDA SAINS KOMPUTER  
DENGAN KEPUJIAN**

**16 FEBRUARI 2003**

---

## ABSTRAK

Sistem Analisis Pemarkahan untuk Penilaian Akademik atau singkatannya **SAP** merupakan sistem yang dicadangkan untuk kegunaan para pensyarah di Fakulti Sains Komputer & Teknologi Maklumat, Universiti Malaya. Ia merupakan satu bentuk sistem sokongan kepada pensyarah yang akan menyediakan perkhidmatan menetapkan gred pelajar secara automatik serta mengeluarkan output dalam bentuk laporan dan graf untuk memudahkan kerja menganalisis pencapaian para pelajarnya.

Laporan ini dibahagikan kepada dua fasa. Fasa pertama merangkumi pengenalan projek, kajian literasi, metodologi & analisis sistem serta rekabentuk sistem yang akan dibangunkan kelak. *Pengenalan* memberikan sedikit gambaran asas terhadap sistem yang bakal dibangunkan. Bab ini memberikan penekanan kepada definisi, objektif, domain/skop dan pelan perancangan pembangunan **SAP**. *Kajian Literasi* pula memberikan huraian ke atas sistem-sistem semasa yang seumpamanya serta rumusan bagi analisis keperluan sistem. Bab ini juga memberikan menyediakan gambaran literasi proses bagi **SAP** termasuk piawaian yang cuba dipenuhinya. *Metodologi dan Analisa Keperluan Sistem* merupakan bab yang terpenting kerana bab ini memberikan huraian terhadap metodologi kajian dan analisis sistem. *Rekabentuk Sistem* pula menyediakan carta alir, carta struktur, struktur pangkalan data dan rekabentuk komponen-komponen sistem yang dicadangkan.

Fasa kedua melibatkan huraian terhadap *Pelaksanaan Sistem*, *Pengujian* dan *Penilaian Sistem*. Tiga bab terakhir akan memberikan maklumat tentang fasa ini. Laporan ini akan diakhiri dengan manual pengguna dan sampel kod aturcara bagi operasi-operasi tertentu.

Diharapkan hasil penulisan laporan ini dapat memberikan manfaat kepada pembaca selain memberi gambaran awal tentang **SAP** sebagai satu sistem sokongan untuk kegunaan para pensyarah.



---

## ABSTRACT

Grading Analysis System for Academic Purpose or its acronym, **SAP**, is a project specially developed for the benefits of instructors at Faculty of Computer Science and Information Technology, University of Malaya. It is a system that provides automatic grading and generate outputs in forms of printable reports and graphs to help them manage their students's academic assessment effectively.

This report consists of two phases . The first one consists of project introduction, literature review and system methodology and design. The first chapter, the *Introduction* deals with the overall view of **SAP**. This chapter covers the system definition, objectives, domain/scope and system development planning. *Literature Review* summarizes some findings of currently implemented system that will help the construction of the proposed system . This part also gives an overview on process design which also includes a list of standards that **SAP** has set to achieve. *System Methodology and Analysis*; the most important part of this report deals with system methodology and analysis required for the system development. *System Design* gives flowcharts, diagrams, database structure and component design that will build the future system.

The second phase which composed of system implementation, testing and system assesment will be the focus of the next three chapters . User manual and some sample on programming codes will also be included at the end of this report.

The author hopes that the report will give some benefits to her readers as well as promoting **SAP** as a supporting tool for instructors.

---

## **PENGHARGAAN**

Penghargaan yang pertama sekali ditujukan kepada Penyelia saya yang budiman, En.Zaidi atas nasihat-nasihat dan penyeliaan yang terbaik. Diikuti dengan *moderator*, En.Noorzailly di atas kesudian beliau mengadili laporan dan persembahan projek saya.

Seterusnya, terima kasih yang tak terhingga kepada rakan-rakan seperjuangan yang sama-sama berkongsi ilmu, masalah dan tunjuk-ajar. Semoga kejayaan menjadi milik kita semua.

Buat keluarga di JB, terima kasih atas dorongan dan kata-kata penawar yang diberikan sepanjang masa. Doa anda semua menjadi azimat untuk meneruskan kehidupan kampus yang semakin mencabar pada bulan-bulan terakhir ini.

Akhir sekali, sekalung tahniah kepada pihak pentadbir Fakulti Sains Komputer dan Teknologi Maklumat yang banyak membantu dalam memberikan kemudahan teknikal dan sokongan fizikal dalam menjayakan projek-projek pelajar Tahap Akhir FSKTM.

Semoga projek yang dibangunkan ini dapat memberikan manfaat kepada pihak FSKTM; atau kepada para pensyarah khususnya. Sekian, terima kasih.

MOHADIMOHAMED

KUALA KRAI

22:15 16/02/03



## KANDUNGAN

Abstrak	i
Abstract	ii
Penghargaan	iii
Kandungan	iv
Senarai Gambarajah	viii
Senarai Jadual	x
Senarai Akronim	xi

### **BAB 1 : PENGENALAN** 1

1.1 Apakah <b>SAP</b> ?.....	2
1.2 Objektif <b>SAP</b> .....	4
1.3 Domain <b>SAP</b> .....	5
1.3.1 Skop pengguna dan sistem.....	5
1.4 Jadual Perancangan Pembangunan Sistem.....	6
1.5 <b>SAP</b> dan Kitar Hayat Pembangunan Sistem .....	7

### **BAB 2 : KAJIAN LITERASI** 9

2.1 Apakah 'sistem'?.....	9
2.2 Kajian ke atas sistem sedia ada.....	10
2.2.1 Ed-Soft .....	11
2.2.1.1 <i>Mentor (Teacher System)</i> .....	12
2.2.2 CS 100 Grading Analysis System.....	15
2.2.3 NDSU Computer Grading System.....	23
2.3 Analisa Keseluruhan Sistem.....	27
2.3.1 Rumusan daripada kajian sistem.....	27
2.4 Kualiti-Kualiti yang cuba dipenuhi oleh <b>SAP</b> .....	31
2.5 Gambaran Proses <b>SAP</b> vs sistem semasa.....	32

**BAB 3 : METODOLOGI DAN ANALISIS KEPERLUAN SISTEM 36**

3.1 Pendekatan pembangunan <b>SAP</b> .....	36
3.2 Teknik pengumpulan maklumat.....	39
3.2.1 Perbincangan dengan Penyelia.....	39
3.2.2 Pembacaan.....	39
3.2.3 Kajian di Internet.....	40
3.2.4 Kajian ke atas sistem semasa.....	40
3.3 Analisis Keperluan.....	41
3.3.1 Keperluan Fungsian SAP.....	41
3.3.2 Keperluan Bukan Fungsian.....	45
3.4 Analisis Keperluan SAP.....	46
3.4.1 Visual Basic 6.0.....	47
3.4.2 Microsoft Access 2000.....	49
3.5 Keperluan Masa Pembangunan.....	50
3.5.1 Perkakasan.....	50
3.5.2 Perisian.....	50
3.6 Keperluan Masa Larian.....	51
3.6.1 Perkakasan.....	51
3.6.2 Perisian.....	51

**BAB 4 : REKABENTUK SISTEM 52**

4.1 Rekabentuk sistem.....	52
4.1.1 Rekabentuk Proses.....	52
4.1.2 Carta Alir Proses.....	54
4.2 Struktur Pangkalan Data <b>PD-SAP</b> .....	66
4.3 Gambarajah Aliran Data (DFD).....	72
4.3.1 Gambarajah Aliran Data Bagi <b>SAP</b> .....	73
4.4 Gambaran Antaramuka Pengguna <b>SAP</b> .....	76



---

**BAB 5 : PELAKSANAAN SISTEM** **78**

5.1	Persekitaran Perkakasan dan Perisian.....	78
5.2	Pembangunan Pangkalan Data.....	79
5.3	Pembangunan Antaramuka Pengguna.....	79
5.4	Pengaturcaraan.....	80
5.5	SAP sebagai Aplikasi.....	82

**BAB 6 : PENGUJIAN SISTEM** **83**

6.1	Pengujian Tahap demi Tahap.....	83
6.1.1	Pengujian Kata Laluan.....	83
6.1.2	Pengujian Kod Kursus.....	84
6.1.3	Pengujian Pangkalan Data.....	84
6.1.4	Pengujian Penjanaan Gred.....	84
6.1.5	Pengujian Penjanaan Graf.....	84
6.1.6	Pengujian Laporan.....	85
6.2	Penyelenggaraan Sistem.....	85

**BAB 7 : PENILAIAN SISTEM** **86**

7.1	Penilaian Keseluruhan Sistem.....	86
7.2	Masalah-Masalah yang Dihadapi.....	86
7.2.1	Masalah Pengaturcaraan.....	87
7.2.2	Perisian-Perisian Tidak Serasi.....	87
7.3	Kelebihan dan Kelemahan Sistem.....	88
7.3.1	Kelebihan Sistem.....	88
7.3.2	Kelemahan Sistem.....	88
7.4	Kesimpulan.....	89

<b>SENARAI RUJUKAN.....</b>	<b>90</b>
-----------------------------	-----------

<b>LAMPIRAN A : DIAGRAM ANAK DALAM DIAGRAM '0'.....</b>	<b>92</b>
---	-----------

<b>LAMPIRAN B : PERGERAKAN BORANG DALAM SAP.....</b>	<b>96</b>
--	-----------

<b>MANUAL PENGGUNA.....</b>	<b>98</b>
-----------------------------	-----------



## SENARAI GAMBARAJAH

### Rajah

1.2	SDLC	7
2.1	Perkaitan di antara Input, Output, Sistem dan Matlamat	10
2.2	Carta Alir Proses bagi modul <i>Mentor</i>	14
2.3	Perincian CS 100-GAS	18
2.4	Sebahagian daripada kamus data CS 100-GAS	20
2.5	Carta Nassi-Schneidermann untuk menunjukkan aliran proses dari Modul 1-5	21
2.6	Contoh perincian modul dan kod pseudo modul	22
2.7	Contoh laporan maklumat pensyarah beserta lembaran kunci	25
2.8	Contoh laporan keputusan pelajar (pilihan)	26
2.9	Gambaran proses semasa	32
2.10	Gambaran proses <b>SAP</b>	34
2.11	Output yang dijangkakan daripada <b>SAP</b>	35
3.1	Model air terjun	38
3.2	Gambarajah pemodulan <b>SAP</b>	44
4.1	Carta Struktur <b>SAP</b>	52
4.2	Pergerakan menu dalam <b>SAP</b>	55
4.3	Pendaftaran masuk ke dalam <b>SAP</b>	56
4.4	Capaian muka ke Menu Utama selepas pengesahan nama pengguna dan katalaluan	57
4.5	Operasi menambah rekod <i>Kursus</i> baru	58
4.6	Operasi menghapus rekod <i>Kursus</i>	59
4.7	Operasi menambah rekod dalam <i>Pelajar</i>	60

4.8	Operasi menghapus rekod dalam <i>Pelajar</i>	61
4.9	Operasi menambah atau menghapus rekod dalam <i>KepUjian</i>	62
4.10	Operasi menambah atau menghapus rekod dalam <i>KepPep</i>	63
4.11	Operasi mencetak output akhir dari <i>KepUjian</i>	64
4.12	Operasi mencetak output akhir dari <i>KepPep</i>	65
4.13	Struktur fail <b>SAP</b>	66
4.14	Medan-medan dalam setiap fail dalam <b>SAP</b>	67
4.15	Diagram konteks <b>SAP</b>	73
4.16	Diagram '0' <b>SAP</b>	74
4.17	Rekabentuk antaramuka <b>SAP</b>	77



---

## SENARAI JADUAL

### Jadual

1.1	Jadual Pembangunan Sistem	6
2.1	Kualiti-Kualiti yang cuba dicapai oleh <b>SAP</b>	31
4.1	Simbol dalam carta alir dan penerangannya	54
4.2	Kamus data bagi <b>PD-SAP</b>	68
4.3	Simbol dalam gambarajah aliran data	72

# SISTEM ANALISIS PEMARKAHAN PENILAIAN AKADEMIK (SAP)

## BAB I: PENDAHULUAN

### SENARAI AKRONIM

<b>SDLC</b>	<i>System Development Life Cycle</i> (Kitar Hayat Pembangunan Sistem).
<b>SAP</b>	Sistem Analisis Pemarkahan untuk Penilaian Akademik.
<b>SAD</b>	<i>System Analysis and Design</i> (Analisis dan Rekabentuk Sistem).
<b>UM</b>	Universiti Malaya
<b>NDSU</b>	North Dakota State University
<b>FTP</b>	<i>File Transfer Protocol</i>
<b>CS 100-GAS</b>	Computer Survival 100 - Grading Automation System
<b>GUI</b>	<i>Graphical User Interface</i> (Antaramuka Pengguna Bergrafik)
<b>VB</b>	Visual Basic
<b>BASIC</b>	<i>Beginner's All Purpose Symbolic Instruction Code</i>
<b>API</b>	<i>Application Programming Interface</i>



---

# SISTEM ANALISIS PEMARKAHAN PENILAIAN AKADEMIK ( **SAP** )

## BAB 1 : PENGENALAN

Sebagai salah satu usaha untuk mempertingkatkan kecekapan dan menaikkan taraf perkhidmatan pengurusan, Universiti Malaya telah banyak melakukan perubahan dalam corak operasinya. Perubahan paling ketara yang dapat dilihat adalah dalam penggunaan sistem perkomputeran dalam sebahagian besar operasinya.

Pembaharuan lain yang tidak kurang pentingnya adalah pengenalan sistem giliran ber kupon di pejabat-pejabat perkhidmatan seperti di bahagian Hal Ehwal Pelajar dan Pejabat Akaun Bendahari. Ia bukan sahaja dapat melicinkan lagi tugas-tugas mereka tetapi juga dapat mempertingkatkan mutu perkhidmatan yang ditawarkan kepada pelajar/pelawat.

Di peringkat Fakulti Sains Komputer dan Teknologi Maklumat (FSKTM) sendiri, selain daripada sistem pendaftaran kursus secara dalam talian yang telah 2 tahun dijalankan, pihak Fakulti juga giat melaksanakan program-program khas seperti projek E-Fakulti untuk menjadikan pentadbiran Fakulti ini 'elektronik' sepenuhnya. E-Fakulti, disertai oleh pelbagai projek lain yang melibatkan pembangunan sistem dan projek multimedia untuk kegunaan akademik, dibangunkan dengan kerjasama pelajar Tahap Akhir melalui projek Latihan Ilmiah menjadi syarat penganugerahan Ijazah Sarjana Muda bagi pelajar-pelajar Sains Komputer dan Teknologi Maklumat.

Sebagai memenuhi aspirasi itu, Sistem Analisis Pemarkahan untuk Penilaian Akademik atau singkatannya **SAP** dibangunkan bagi projek Latihan Ilmiah I ini sebagai salah satu sistem sokongan bagi kegunaan para pensyarah FSKTM.



---

## 1.1 APAKAH **SAP** ?

Pada masa ini, penentuan gred dilakukan secara manual. Apabila seseorang pensyarah selesai memeriksa kertas peperiksaan pelajar, penetapan gred turut dilakukan secara manual. Seterusnya, setiap pensyarah perlu melaporkan setiap keputusan gred pelajar ke pejabat FSKTM kerana satu-satunya sistem yang menyimpan keputusan peperiksaan pelajar terdapat di situ. Pensyarah terpaksa meninggalkan bilik mereka dan pergi ke pejabat untuk proses masukan data gred berkenaan.

Sistem Analisis Pemarkahan untuk Penilaian Akademik atau **SAP** adalah satu sistem bersifat *stand-alone* bagi kegunaan pensyarah untuk menentukan gred yang diperolehi secara automatik, menyimpan data-data berkenaan, menganalisis keputusan pelajar bagi setiap kursus berkenaan untuk mengesan pelajar-pelajar bermasalah. dan menyediakan laporan gred pelajar ke pejabat

Berikut adalah perincian perkhidmatan yang ditawarkan oleh **SAP** :

- 1) Apabila pengguna (pensyarah) menginput markah yang diperolehi oleh setiap pelajar bagi satu-satu kertas ujian dan peperiksaan untuk setiap kursus berkenaan, sistem akan melakukan pemprosesan untuk menjanakan gred yang berpadanan dan gred itu akan dipaparkan kepada pengguna.
- 2) Semua maklumat gred pelajar bagi setiap kursus berkenaan akan disimpan dalam satu pangkalan data yang dikenali sebagai **PD-SAP**..
- 3) Analisis akan dibuat ke atas keputusan gred pelajar setiap kursus untuk menilai keberkesanan kursus dan analisis *benchmarking* untuk mengenalpasti kelompok pelajar yang memerlukan bimbingan.
- 4) Hasil analisis akan dipaparkan dalam bentuk graf dan laporan gred dalam format tertentu disediakan untuk simpanan pejabat FSKTM.



---

Secara umumnya, **SAP** melibatkan dua mekanisme analisis; iaitu semasa memberikan gred yang berpadanan bagi setiap markah yang diinputkan dan semasa menganalisis data gred untuk memaparkan graf. **SAP** turut melibatkan mekanisme menginput, menyimpan, mengemasikini dan mencapai kembali data-data gred pelajar yang disimpan dalam pangkalan datanya.

Sistem ini diharapkan dapat memudahkan kerja-kerja para pensyarah dari aspek:

- 1) Memudahkan tugas penetapan gred ke atas setiap pelajar kerana sistem akan melaksanakannya dengan lebih berkesan.
- 2) Menyimpan data-data gred pelajarnya mengikut kursus-kursus kendaliannya secara lebih sistematik
- 3) Membolehkan para pensyarah menganalisis keputusan pelajarnya dengan lebih berkesan melalui analisis bergraf dan *benchmarking*.
- 4) Membolehkan para pensyarah melibatkan diri dengan lebih aktif sebagai pemudahcara (*facilitator*) dalam membimbing pelajar-pelajar yang bermasalah dalam kusus-kursus berkenaan.
- 5) Hasil analisis itu juga membolehkan pensyarah merangka program peningkatan mutu kursus kendaliannya seperti menggubal kandungan kursus dan mempelbagaikan teknik pengajaran.
- 6) Menyediakan laporan keputusan pelajar mengikut format tertentu yang boleh diterim oleh pejabat FSKTM.

Sistem ini hanya akan digunakan oleh para pensyarah, bersifat *stand-alone* (tidak dihubungkan ke rangkaian dalam talian) dan akan dibangunkan menggunakan pendekatan antaramuka muka pengguna yang mudah dan pangkalan data yang sistematik.



---

## 1.2 OBJEKTIF **SAP**

Matlamat **SAP** ialah untuk mempermudah tugas menginput, menentukan gred, menyimpan, mengemaskini, mencapai dan menganalisis data gred berkenaan untuk merancang program kaunseling, penyediaan laporan dan merancang semula program peningkatan kursus.

Beberapa objektif yang lebih spesifik telah digariskan untuk mencapai matlamat tersebut. **SAP** harus berupaya:

- 1) Untuk menentukan gred yang berpadanan secara automatik bagi setiap markah ujian/peperiksaan yang diinputkan oleh pensyarah bagi setiap pelajarinya bagi setiap kursus kendaliaannya.
- 2) Untuk menyimpan maklumat gred berkenaan ke dalam pangkalan data khas yang turut menyimpan data-data lain tentang setiap pelajar kursus berkenaan.
- 3) Untuk menganalisis keputusan gred setiap kursus dan mengeluarkan output analisis dalam bentuk graf yang lebih mudah difahami.
- 4) Untuk menghasilkan laporan gred peperiksaan pelajar dalam format yang boleh diterima oleh pihak pentadbir FSKTM (samada bentuk pindahan fail atau laporan bertulis).

Apabila keempat-empat objektif ini dapat dipenuhi, **SAP** bukan sahaja dapat menyediakan satu sistem sokongan yang cekap kepada pensyarah tetapi juga membantu mempertingkatkan potensi mereka sebagai penasihat kepada para pelajar sekaligus mempertingkatkan mutu pengajaran mereka.

## 1.3 DOMAIN **SAP**



---

Domain sistem ini adalah 'gred' ujian dan peperiksaan bagi setiap pelajar yang mendaftar bagi setiap kursus yang dikendalikan oleh seseorang pensyarah. Gred adalah hasil daripada mekanisme analisis pertama iaitu apabila markah ujian dan peperiksaan diinputkan, sistem akan menentukan gred yang berpadanan. Dari set gred inilah, mekanisme analisis kedua dilaksanakan iaitu untuk menjanakan graf yang membolehkan *benchmarking* dibuat untuk membolehkan pencapaian pelajar diselidiki dan keberkesanan kursus ditentukan. Pemarkahan ujian dan peperiksaan merupakan satu rutin kepada para pensyarah; di mana mereka mengendalikan pemarkahan untuk lebih daripada satu kursus pada satu-satu semester. Maka, penggredan merupakan apa yang cuba dilakukan oleh **SAP**.

Selain itu, sistem ini hanya akan berada dalam lingkungan pejabat satu-satu pensyarah sahaja. Sebagai sebuah sistem yang bersifat *stand-alone*, ia tidak dihubungkan ke rangkaian LAN FSKTM atau mana-mana rangkaian atas talian. Ia dibangunkan semata-mata untuk kegunaan pensyarah untuk memudahkan mereka menentukan gred pelajar, menyimpan data gred berkenaan dan menganalisis gred berkenaan untuk membantu mereka merancang tindakan susulan. Oleh itu, seseorang pensyarah hanya akan dapat mengendalikan dan mencapai keputusan para pelajar di bawah kursus kendaliannya sahaja.

### 1.3.1 SKOP PENGGUNA DAN SISTEM

Seperti yang telah dijelaskan sebelumnya, sistem ini dibangunkan untuk kegunaan para pensyarah. Oleh itu, pensyarah merupakan satu-satunya pengguna bagi sistem ini. Skop tugas pensyarah sebagai pengguna sistem ini ialah menginputkan markah bagi ujian dan peperiksaan pelajar, menyimpan gred yang ditetapkan oleh sistem secara automatik ke dalam **PD-SAP**, mengemaskini sebarang perubahan data dan mengarahkan penghasilan gred dan laporan gred pelajar samada dalam bentuk bertulis ataupun pindahan fail.

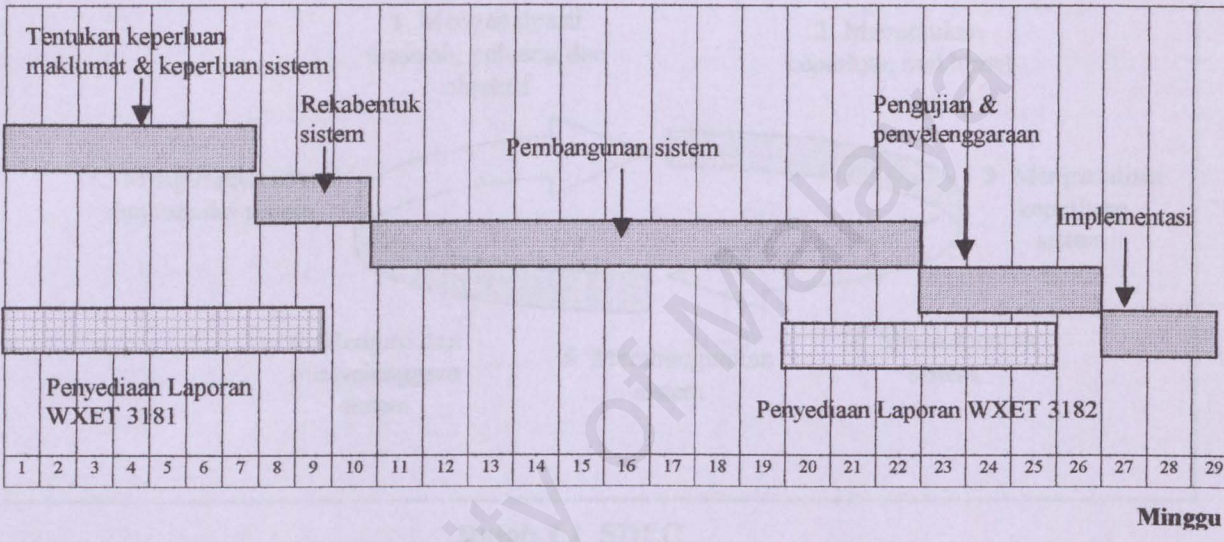


Skop sistem pula ialah menjana gred bagi setiap input markah, menyimpan data gred secara automatik dan sistematik, menjana graf sebagai output analisis gred dan menghasilkan laporan gred menurut kehendak pengguna sistem.

1.4 JADUAL PERANCANGAN PEMBANGUNAN SAP

Perancangan pembangunan sistem ini dapat digambarkan oleh gambarajah berikut:

Aktiviti

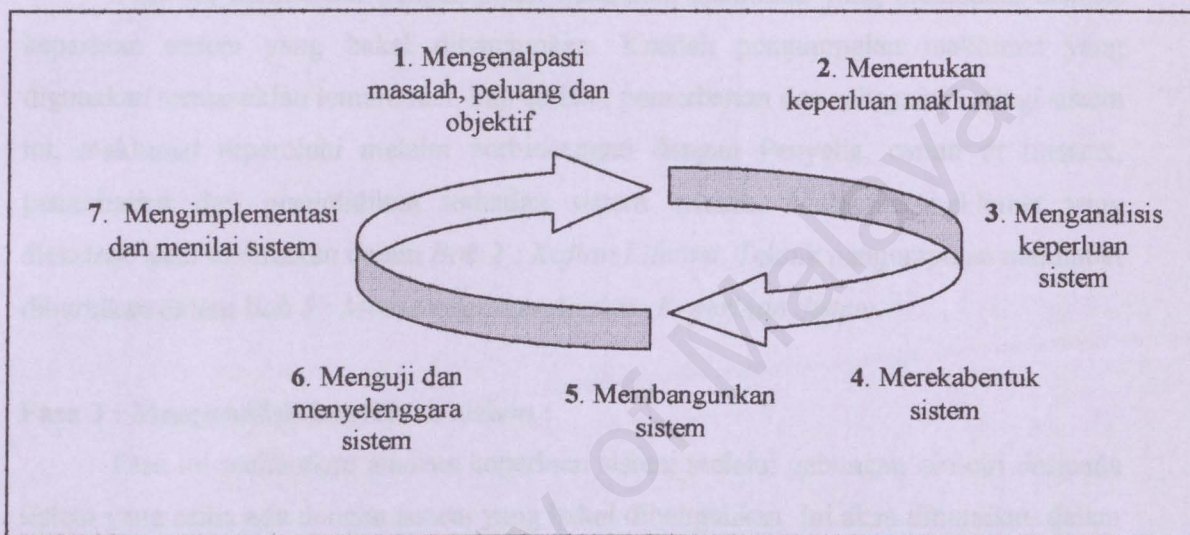


Jadual 1.1 : Jadual Pembangunan Sistem



## 1.5 KITAR HAYAT PEMBANGUNAN SISTEM

**SAP** dibangunkan mengikut model Kitar Hayat Pembangunan Sistem atau SDLC (*System Development Life Cycle*). SDLC merupakan satu pendekatan berfasa untuk menganalisis dan merekabentuk sesuatu sistem mengikut satu kitar yang berpadanan dengan objektif pembangun dan aktiviti pengguna (Kendall dan Kendall 1999: 7). Ia terdiri daripada tujuh fasa seperti yang ditunjukkan dalam rajah 1.2 :



**Rajah 1.1 SDLC**

Bagi Latihan Ilmiah I , hanya fasa pertama hingga keempat sahaja yang akan terlibat. Fasa kelima hingga ketujuh akan diteruskan bagi Latihan Ilmiah II. Berikut adalah perincian bagaimana pembangunan sistem ini berkait dengan SDLC:

### **Fasa 1 : Mengenalpasti masalah, peluang dan objektif:**



---

Fasa ini melibatkan pemahaman terhadap masalah yang dihadapi oleh organisasi dan peluang yang melibatkan sesi temuramah dengan bakal pengguna sistem, merumuskan penemuan-penemuan penting daripada pertemuan itu dan mula menentukan skop projek sistem yang akan dibangunkan. Seterusnya, hasil kajian didokumentasikan. Pemahaman masalah, mengenalpasti peluang yang ada dan merancang objektif telah pun dihuraikan dengan terperinci dalam *Bab 1: Pengenalan*.

#### **Fasa 2 : Menentukan keperluan maklumat :**

Fasa ini melibatkan pengumpulan maklumat-maklumat yang membantu analisis keperluan sistem yang bakal dibangunkan. Kaedah pengumpulan maklumat yang digunakan termasuklah temuramah, kaji selidik, pemerhatian dan sebagainya. Bagi sistem ini, maklumat diperolehi melalui perbincangan dengan Penyelia, carian di Internet, pemerhatian dan penyelidikan terhadap sistem semasa. Maklumat-maklumat yang diekstrak akan dihuraikan dalam *Bab 2 : Kajian Literasi*. Teknik pengumpulan maklumat dihuraikan dalam *Bab 3 : Metodologi dan Analisis Keperluan Sistem*.

#### **Fasa 3 : Menganalisis keperluan sistem :**

Fasa ini melibatkan analisis keperluan sistem melalui gabungan ciri-ciri daripada sistem yang sedia ada dengan sistem yang bakal dibangunkan. Ini akan dihuraikan dalam *Bab 3 : Metodologi dan Analisis Keperluan Sistem*.

#### **Fasa 4 : Menyediakan rekabentuk sistem :**

Fasa ini melibatkan proses rekabentuk sistem secara logikal seperti rekabentuk masukan data, rekabentuk borang yang efektif, rekabentuk fail dan pangkalan data serta prosedur kawalan dan 'back-up'). Huraian adalah dalam bentuk gambaran DFD untuk menggambarkan input, proses dan output, kamus data untuk pangkalan data dan carta alir proses untuk keputusan berstruktur. Ini akan dihuraikan dalam *Bab 4: Rekabentuk Sistem*.

Tiga fasa terakhir yang merangkumi pelaksanaan, pengujian dan penilaian akan diterangkan dalam *Bab 5, 6 dan 7*.



## BAB 2 : KAJIAN LITERASI

Bab ini dibahagikan kepada 5 topik:

2.1 Apakah 'sistem'?

2.2 Kajian ke atas sistem sedia ada.

2.3 Analisis Keseluruhan Sistem.

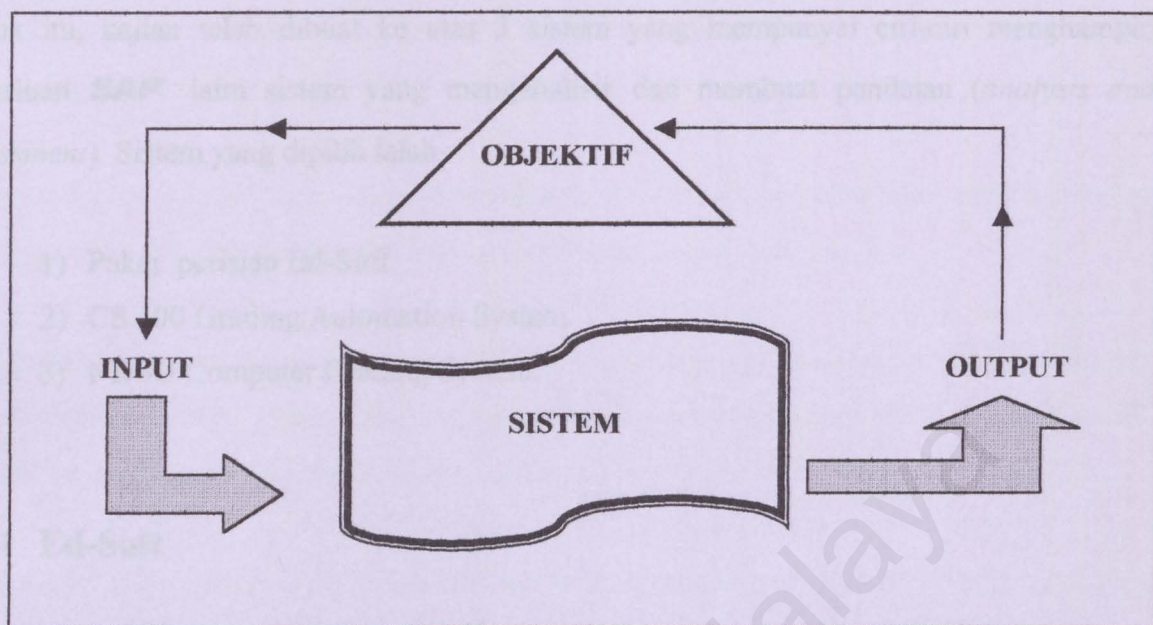
**2.4** Kualiti-kualiti yang cuba dipenuhi oleh **SAP**

2.5 Gambaran proses **SAP** vs sistem semasa.

### 2.1 APAKAH 'SISTEM'?

Menurut Ian Sommerville, sistem bermaksud sekumpulan komponen yang saling berkait dan bekerjasama untuk mencapai objektif-objektif tertentu (Sommerville, 1995). Komponen-komponen itu boleh jadi satu komponen yang boleh berdiri sendiri; sesetengahnya perlu dihubungkan dengan komponen lain untuk menjalankan fungsinya. Komponen ini juga disebut sebagai subsistem, kerana ia boleh jadi sebuah sistem. Kejayaan setiap komponen berfungsi dalam satu-satu sistem bergantung kepada komponen-komponen lain. Sebagai contoh, sesuatu perisian sistem hanya boleh beroperasi apabila prosessor berfungsi. Prosessor pula hanya boleh menjalankan pengiraan jika sistem perisian dapat mengesahkan pengiraan ini berjaya dipasang pada sistem.

Semua sistem memproses input daripada persekitaran dan menghasilkan output. Proses-proses yang biasa termasuklah proses mengemaskini, mencetak dan mengira. Output daripada sistem merupakan satu penunjuk yang baik untuk menentukan samada sistem tersebut berjaya mencapai objektif atau tidak (rujuk Rajah 2.1).



(Kendall dan Kendall 1996: 29)

**Rajah 2.1** Perkaitan di antara input, output, sistem dan matlamat.

Bagi **SAP**, input ke dalam sistem ialah 'markah' yang akan dimasukkan oleh pensyarah bagi setiap pelajar, sistem akan melakukan pemprosesan dan mengeluarkan 'gred' sebagai output. Selain daripada 'gred', output juga dihasilkan dalam bentuk 'graf' bagi keputusan setiap kursus dan 'laporan' untuk dihantar ke Pejabat FSKTM.

## 2.2 KAJIAN KE ATAS SISTEM SEDIA ADA

Untuk lebih memahami bagaimana sesuatu sistem itu seharusnya dibangunkan, kajian perlu dibuat ke atas sistem yang sedia ada, sama ada sistem yang serupa dengannya atau hampir menyamai sistem tersebut. Daripada sistem-sistem ini, maklumat tentang keperluan sistem yang bakal dibangunkan dapat ditentukan dengan mudah. Selain itu, kajian ke atas kebaikan dan kelemahan sistem-sistem berkenaan membuka ruang kepada pembangunan sistem baru yang lebih baik.



Untuk itu, kajian telah dibuat ke atas 3 sistem yang mempunyai ciri-ciri menghampiri keperluan **SAP** iaitu sistem yang menganalisis dan membuat penilaian (*analysis and assessment*). Sistem yang dipilih ialah :

- 1) Pakej perisian Ed-Soft.
- 2) CS 100 Grading Automation System.
- 3) NDSU Computer Grading System.

### 2.2.1 Ed-Soft

Ed-Soft ialah satu pakej perisian keluaran Ed-Soft Software Corporation yang berpusat di Dallas, Texas, USA ([www.ed-soft.com](http://www.ed-soft.com) : 2/7/2001). Ed-Soft ialah satu aplikasi perisian bersepadu yang meningkatkan keberkesanan dan kecekapan pendidik dalam memantau dan memenuhi keperluan setiap individu pelajarinya setiap hari. Ia dibangunkan berdasarkan pembangunan kurikulum pembelajaran yang asas. Sebagai satu pakej aplikasi yang berorientasikan web, ibu bapa di rumah boleh mencapai rekod pencapaian dan penilaian anak-anak mereka serta maklumat tenaga pengajar, sekolah atau universiti.

Ciri nyata pada Ed-Soft yang hampir sama dengan **SAP** ialah pakej aplikasi memperlihatkan pengukuran dan penanda (*indicator*) prestasi pembelajaran pelajar. Ed-Soft turut berkongsi matlamat yang hampir sama dengan **SAP** iaitu membolehkan tenaga pengajar menjimatkan masa dalam hal-hal pentadbiran dan menumpukan masa kepada peningkatan teknik pembelajaran dan menjadi penasihat yang aktif kepada para pelajarinya.

Ed-Soft dibangunkan menggunakan pendekatan *state-of-the-art* yang terdiri daripada enam modul iaitu :



- 1) *Online Benchmarking Testing* - bersifat *stand-alone*, penilaian pelajar dibuat berdasarkan objektif dan maklum balas prestasi pelajar dibuat secara segera.
- 2) *Foundation (Curriculum System)* - membantu rekabentuk, pembangunan, pengurusan dan pengagihan kurikulum asas dan kandungan kursus secara bersepadu.
- 3) *Mentor (Teacher System)* - membolehkan rekabentuk, pembangunan, pengurusan dan penghantaran kand. kursus asas termasuk sukatan pembelajaran, aktiviti pembelajaran, rancangan kursus dan penilaian pelajar secara efektif.
- 4) *Achieve (Student System)*- menyediakan pengagihan tugas yang spesifik mengikut prestasi pelajar secara online termasuk penilaian tugas dan penilaian *benchmark*.
- 5) *Engage (Parent System)* – membolehkan capaian selamat ibu bapa ke atas data-data pelajar, menyediakan analisis prestasi dan laporan daripada pengajar.
- 6) *Encompass (District Administration System)* - menyediakan maklumat pengurusan dan laporan data-data pentadbiran dan demografi berkaitan tenaga-tenaga pengajar, kelas-kelas, pelajar, kursus-kursus, kampus, jadual kelas, kelulusan dan sebagainya.

Walaupun bagaimanapun, kajian literasi ini hanya akan memfokus kepada modul *Mentor* kerana ia berkait rapat dengan tugas-tugas pensyarah terutama dalam aspek mengendalikan data-data pelajar dan rekod prestasi mereka.

### **2.2.1.1 Mentor (Teacher System)**

Modul *Mentor* memberikan penyelesaian praktikal dan keupayaan berteknologi untuk membantu dan mempermudah aktiviti seharian tenaga pengajar, menguruskan dan



merancang kelas, menyediakan tugas, mengumpulkan dan menilai tugas pelajar, serta menyediakan penilaian prestasi pelajar secara dramatik. Penilaian pelajar dapat dilihat melalui *Classroom Dashboard*, satu ciri yang ditawarkan oleh sistem yang menggunakan pakej aplikasi ini.

Berikut adalah kelebihan *Mentor*:

- 1) memudahkan pengajar melihat prestasi pelajar dan membuat penilaian prestasi pelajar
- 2) memudahkan pengajar mengenalpasti kelompok pelajar yang memerlukan kaunseling dan bimbingan pelajaran yang lebih spesifik
- 3) menjimatkan masa tenaga pengajar dalam tugas-tugas pentadbiran (seperti merekod data keputusan pelajar secara manual) untuk memberi lebih banyak tumpuan ke atas memperbaiki teknik pengajaran dan mengawasi perkembangan pelajar (*mentoring*).
- 4) Merealisasikan matlamat IT dalam pengurusan dan pendidikan iaitu 'maklumat di hujung jari'.

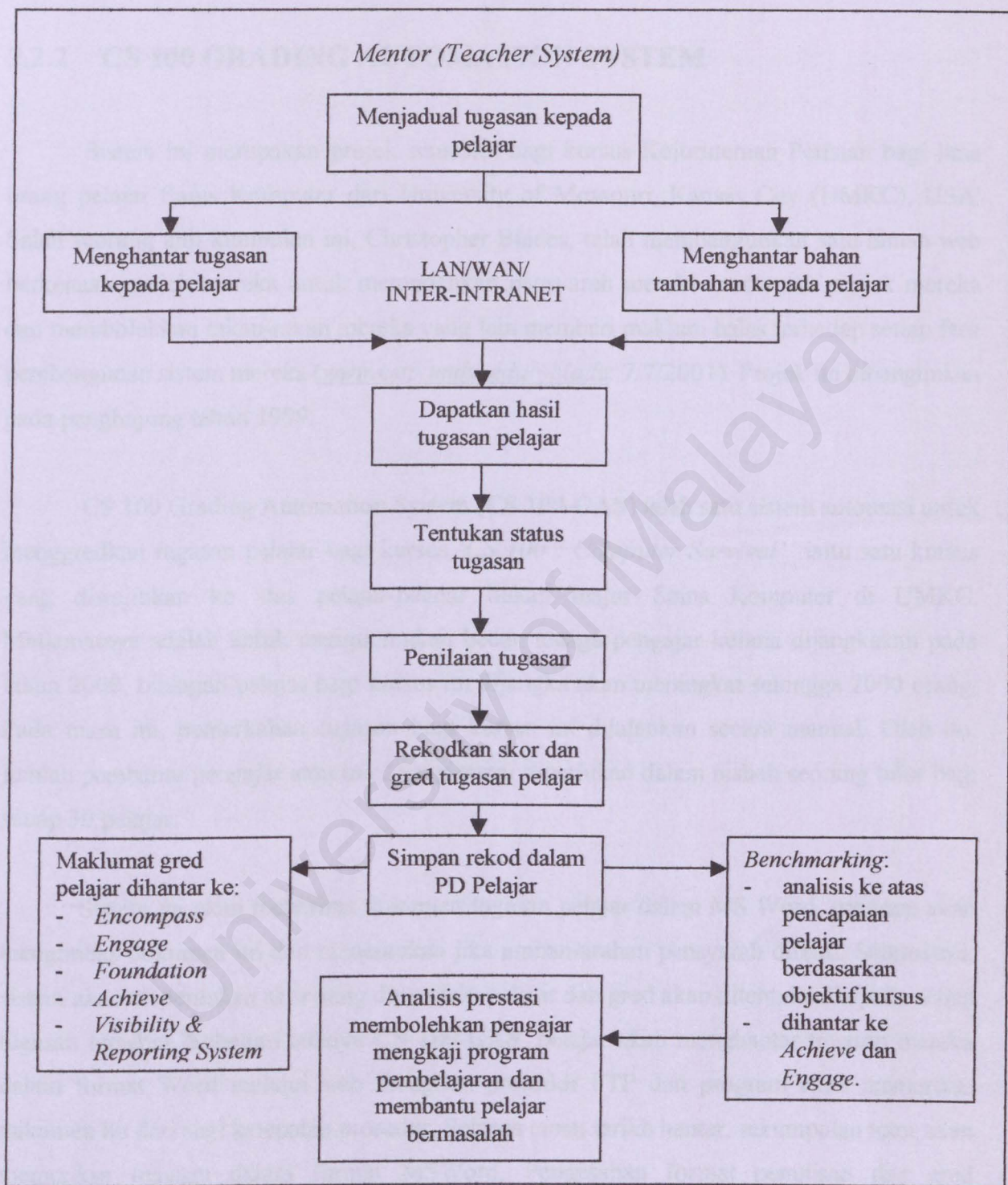
Walau bagaimanapun, ia masih mempunyai beberapa kelemahan. Oleh kerana ia didatangkan dalam satu pakej bersepadu, ia mungkin tidak menghadapi masalah besar, cuma ia mungkin membataskan kreativiti pengajar dalam merekabentuk sistem simpanan maklumat pelajar yang menurut keperluannya. Ini adalah kelemahan biasa apabila menggunakan perisian *generic* yang khusus (*fixed*) (Sommerville, 1995).

Selain itu, *Mentor* amat bergantung kepada keupayaan rangkaian LAN/WAN. Jika rangkaian tersebut menghadapi masalah (seperti masalah pelayan (*server*), virus, *software corrupt*, capaian lambat sebagainya), bukan sahaja gerak kerja pensyarah, malah keutuhan data dan sistem akan terganggu. Ini adalah cabaran biasa dalam membangunkan aplikasi-aplikasi yang saling berkait dan bergantung.

Gambaran proses dalam modul *Mentor* dapat dihuraikan oleh gambarajah berikut:



### Mentor (Teacher System)



**Rajah 2.2 Modul Mentor**



### 2.2.2 CS 100 GRADING AUTOMATION SYSTEM

Sistem ini merupakan projek semester bagi kursus Kejuruteraan Perisian bagi lima orang pelajar Sains Komputer dari University of Missouri, Kansas City (UMKC), USA. Salah seorang ahli kumpulan ini, Christopher Blades, telah membangunkan satu laman web berkenaan projek mereka untuk memudahkan pensyarah mereka memantau projek mereka dan membolehkan rakan-rakan mereka yang lain memberi maklum balas terhadap setiap fasa pembangunan sistem mereka ([www.cstp.umkc.edu/~blade](http://www.cstp.umkc.edu/~blade): 7/7/2001). Projek ini dibangunkan pada penghujung tahun 1999.

CS 100 Grading Automation System (**CS 100-GAS**) ialah satu sistem automasi untuk menggredkan tugas pelajar bagi kursus '*CS 100 : Computer Survival*' iaitu satu kursus yang diwajibkan ke atas pelajar-pelajar bukan major Sains Komputer di UMKC. Matlamatnya adalah untuk mengurangkan beban tenaga pengajar kerana dijangkakan pada tahun 2000, bilangan pelajar bagi kursus ini dijangka akan meningkat sehingga 2000 orang. Pada masa ini, pemarkahan tugas bagi kursus ini dijalankan secara manual. Oleh itu, jumlah pembantu pengajar atau tutor yang ramai diperlukan dalam nisbah seorang tutor bagi setiap 30 pelajar.

Sistem ini akan menerima dokumen tugas pelajar dalam MS Word, program akan mengimbas dokumen itu dan menentukan jika arahan-arahan pensyarah diikuti. Seterusnya, sistem akan menentukan skor yang diperolehi pelajar dan gred akan ditentukan kepada setiap tugas tersebut. Sebelum adanya **CS 100-GAS**, pelajar akan menghantar tugas mereka dalam format Word melalui web mengikut prosedur FTP dan program akan memeriksa dokumen itu dari segi ketepatan prosedur. Selepas tamat tarikh hantar, sekumpulan tutor akan memeriksa tugas dalam format MSWord. Pengesahan format penulisan dan gred dijalankan secara manual oleh pensyarah.



Pembangunan sistem melibatkan semua fasa pembangunan sistem termasuklah dokumentasi spesifikasi dan keperluan *preliminary*, prototaip, pengujian dan implementasi dan penyediaan manual pengguna. Dokumentasi sistem dibahagikan kepada dokumentasi dalaman (keperluan dan spesifikasi sistem) dan dokumentasi pengguna (gambaran rekabentuk sistem).

Sistem dibangunkan menggunakan MSAccess dan bahasa pengaturcaraan C++ di atas platform Windows 95 atau NT. Beberapa komputer peribadi dalam lingkungan pejabat CSTP dihubungkan ke rangkaian UMKC CSTP (Fakulti terbabit). Setiap komputer berkenaan mempunyai capaian terus dari rangkaian NT dan 'cluster' VMS kampus. Sistem akan melarikan aplikasi MSWord 97 manakala 'cluster' VMS untuk mencapai tugas yang dihantar.

Sistem ini mempunyai beberapa kekangan. **CS 100-GAS** mesti boleh mengekstrak kod format dokumen dalam format MSWord. Untuk itu, pembangun sistem ini perlu memahami penggunaan MFC (MS Foundation Classes) untuk memahami simbol-simbol yang dijanakan dalam dokumen MSWord. Mereka menjalani satu sesi latihan khas daripada pakar teknikal dari Microsoft Corporation.

Selain itu, sistem mesti mampu menerima input yang berjujukan sehingga 2000 dokumen (kerana dianggarkan terdapat 2000 pelajar). Semasa operasi, parser<sup>1</sup> mesti mengekalkan kawalan supaya program menerima tugas yang dihantar secara berterusan semasa waktu bukan puncak tanpa penyeliaan. Parser juga mesti sentiasa sedia untuk pengubahsuaian data oleh pengguna (pensyarah/tutor) untuk menguruskan keperluan penggredan tambahan.

---

<sup>1</sup> Parser ialah sejenis peranti yang berperanan sebagai penyurai atau menapis data-data berskala besar..



**GS 100-GAS** memberikan beberapa faedah yang ketara. Pensyarah dapat menjimatkan masa dalam menilai tugas pelajar kerana penilaian dijalankan secara automatik oleh sistem ini. Oleh itu, mereka dapat menumpukan masa kepada projek-projek pelajar yang jauh lebih penting. Penggredan secara automatik juga dapat mengelakkan *bias* atau pilih kasih. Selain itu, **GS 100-GAS** dapat menjimatkan kos pentadbiran Universiti. Dengan mengurangkan bilangan tutor untuk menilai tugas pelajar bermakna pengurusan dapat menjimatkan belanja caj perkhidmatan. Tanpa **CS 100-GAS**, 67 tutor diperlukan bagi menilai 2000 tugas sedangkan dengan **CS 100-GAS**, lima tutor sudah memadai.

**CS 100-GAS** telah dibangunkan dalam masa 12 minggu. Sistem ini dibahagikan kepada 5 modul<sup>2</sup> untuk memudahkan pembangunannya. Dokumentasi sistem dalam bentuk DFD (*Data-Flow Diagram*), kamus data untuk pangkalan data pelajar, *pseudo code C++* dan rekabentuk antaramuka pengguna telah dilampirkan dengan baik dalam laman web ini untuk membolehkan pensyarah dan rakan-rakan mereka memberi maklumbalas. Satu percubaan yang menarik ialah mereka cuba menggambarkan modul-modul itu menggunakan pendekatan carta Nassi-Scheiderman yang lebih jelas dan mudah difahami.

---

<sup>2</sup> Modul ialah segmen-segmen kod yang menjalankan satu fungsi yang spesifik.



Berikut adalah rajah perincian proses GS 100-GAS:

◆ **Input sistem :**

- ❑ Setiap pelajar diberi satu tugas dalam format MSWord beserta 6 arahan asas untuk menukar kandungan dokumen itu (seperti menukar *indentation*, tukar stail *font* dan sebagainya).
- ❑ Program akan memeriksa samada semua arahan pembetulan dilaksanakan.
- ❑ Skor diberikan antara 5 hingga 10.
- ❑ Gred ditentukan secara automatik oleh sistem.



◆ **'Parsing' secara automatik ke atas tugas pelajar CS 100 :**

- ❑ Rutin 'parsing' dihubungkan kpd mekanisme import-eksport yang bertindak dalam masa nyata untuk mencapai semula tugas itu dan menghantar gred kpd kepada pangkalan data pelajar.
- ❑ Mereka bentuk '*document parser*' menggunakan C++ untuk semakan tugas pelajar. Semakan dibuat melalui perbandingan setiap tugas dengan satu dokumen kunci yg tepat (*template*).
- ❑ Sekali lagi parser diintegrasikan kepada mekanisme import-eksport yang dibangunkan menggunakan C++ untuk penggredan secara *online*.

**Rajah 2.3 Perincian CS 100-GAS**

Program yang dibangunkan bertanggungjawab untuk memeriksa setiap dokumen tugas dengan tepat melalui mekanisme perbandingan yang efektif. Program juga mesti mampu 'menolak' satu markah daripada setiap satu kesilapan dengan skor minimum iaitu lima. Output akhir (skor dan gred) bagi setiap tugas hendaklah disimpan dalam pangkalan data pelajar.



Sebagai sistem yang diintegrasikan kepada UMKC CSTP, **CS 100-GAS** harus mengelakkan aplikasinya bercampur-aduk dengan aplikasi lain. Oleh kerana sistem akan terus berjalan sepanjang masa, sistem akan turut mencapai fail-fail tertentu dalam rangkaian CSTP. Oleh itu, satu mekanisme semakan ralat perlu diwujudkan untuk memastikan hanya fail tertentu sahaja yang boleh dicapai

Keutuhan data juga merupakan satu aspek yang diberikan perhatian. Setiap dokumen MSWord yang dihantar mesti diterima dan disimpan dalam bentuk asalnya tanpa ada pengubahsuaian. Selain itu, gred pelajar mesti dieksport dengan selamat ke pangkalan data. Pengubahsuaian gred secara tidak sah adalah satu risiko yang mungkin berlaku. Walau bagaimanapun, ia bergantung kepada tahap keselamatan sistem CSTP sendiri sebagai rangkaian induknya.

Beberapa keperluan operasi telah digariskan seperti tempoh 'parsing' tidak lebih daripada 1 saat. Modul 'parser' dan modul output akan beroperasi serentak. Pelajar yang ingin mencapai sistem ini dari rumah pula memerlukan kelajuan capaian pada kapasiti sekurang-kurangnya 28.8 kbps. Bahagian input mesti mampu meminimumkan keperluan *bandwidth* untuk menerima tugas yang dihantar dan menghantar keputusan gred kepada pelajar. Oleh itu, penggunaan bunyi dan grafik tidak digunakan di antara antaramuka pensyarah- pelajar.

Selepas minggu ke-14, sistem ini mula diuji oleh sekumpulan pensyarah kursus CS 100. Perisian sistem ini diintegrasikan ke dalam persekitaran UMKC. Sistem ini turut dihubungkan ke laman web kursus CS 100 supaya pelajar boleh membuat capaian ke atas keputusan tugas mereka. Walau bagaimanapun, ujian input tidak melibatkan 2000 input, sebaliknya tidak kurang daripada 10 input sahaja dimasukkan sebagai ujian.



Berdasarkan daripada pengujian yang dilakukan, sistem ini telah berjaya menyemak tugas pelajar dan memberikan skor dan menetapkan gred secara automatik kurang daripada 1 saat. Walau bagaimanapun, terdapat satu kelemahan yang ketara di mana sistem didapati tetap menerima tugas yang dihantar melebihi tarikh akhir penghantaran. Tiada mekanisme khusus untuk mengesan tugas yang terlewat hantar; ini menyebabkan tutor masih meneruskan pengesahan gred secara manual kerana tiada pengesahan status tugas dibuat.

Selain itu, penghantaran maklumat gred ke pangkalan data didapati agak 'lambat'. Oleh itu, beberapa perubahan dicadangkan kepada pasukan pembangunan sistem ini untuk mempertingkatkan keupayaan sistem.

Berikut adalah contoh sebahagian daripada dokumentasi projek CS 100-GAS :

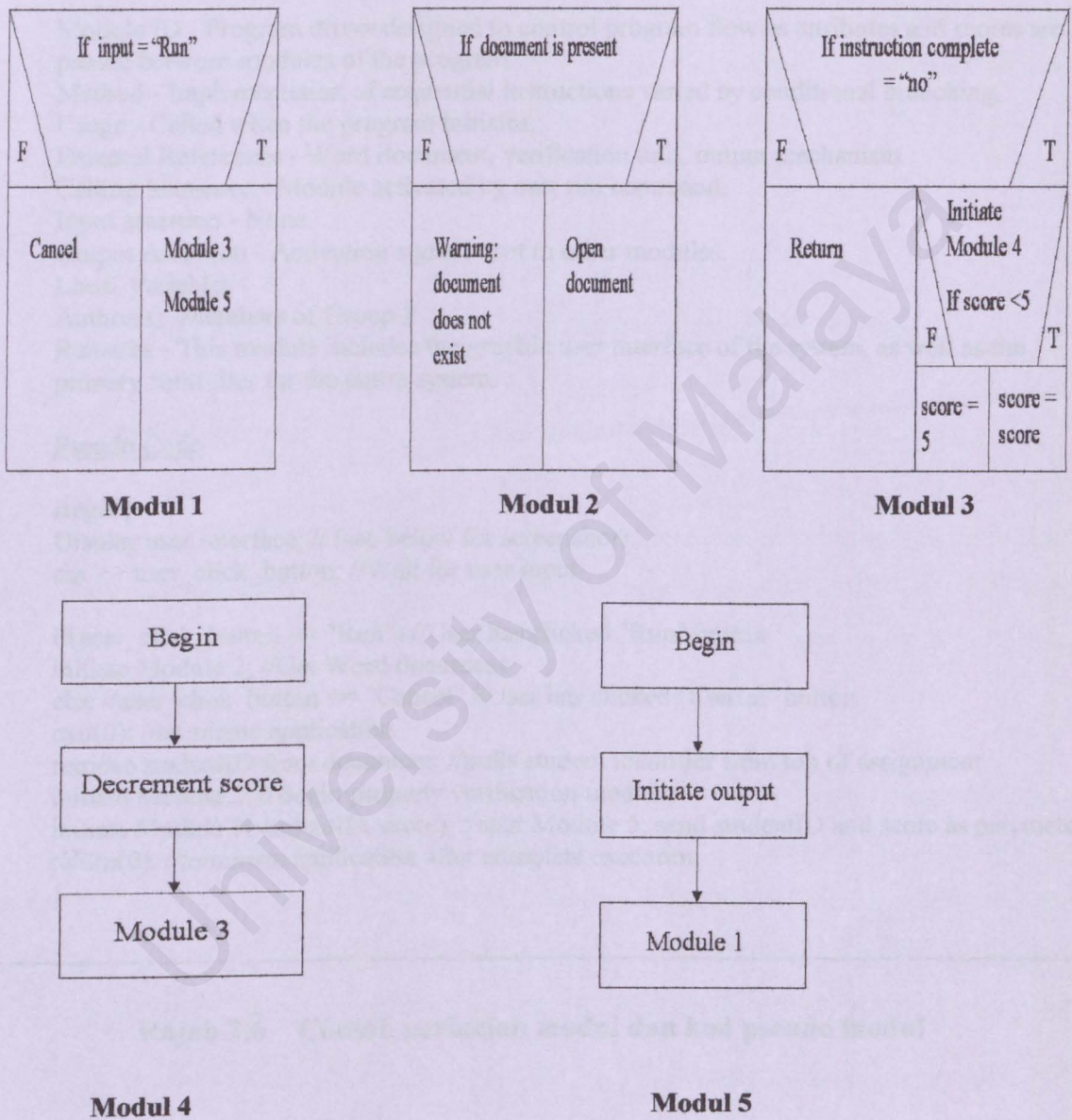
**Data Dictionary**

VARIABLE	DATA TYPE	DESCRIPTION
objWordApp	Application	User-defined class; defines one instance of MSWord
ObjDoc	Document	User-defined class; defines one instance of document in MSWord
objTabStops	TabStops	user-defined collection of tabstops
objTabStop	TabStop	user-defined collection of properties of single tabstop
TabStop_First	float	value of position of first tab stop in paragraph
TabStop_Second	float	value of positionn of second tab stop in paragraph
objWords	Words	user-defined collection of words

**Rajah 2.4 Sebahagian daripada kamus data CS 100-GAS**



**Rajah 2.5 Carta Nassi-Schneidermann untuk menunjukkan aliran proses dari Modul 1 hingga 5.**



Contoh perincian modul 1 dan sebahagian daripada kod pseodo C++nya:

#### Modul 1:

Module ID - Program driver designed to control program flow as attributes and scores are passed between modules of the program.

Method - Implementation of sequential instructions varied by conditional branching.

Usage - Called when the program initiates.

External References - Word document, verification unit, output mechanism.

Calling Sequence - Module activated by user run command.

Input assertion - None.

Output Assertion - Activation signals sent to other modules.

Local Variables -

Author(s) -Members of Group F

Remarks - This module includes the graphic user interface of the system, as well as the primary controller for the entire system.

#### Pseudo Code

Begin{

Display user interface; // (see below for screenshot)

cin >> user\_click\_button; //Wait for user input

if(user\_click\_button == "Run") //User has clicked "Run" button

initiate Module 2; //Get Word document

else //user\_click\_button == "Cancel" //User has clicked "Cancel" button

exit(0); //terminate application

retrieve studentID from document; //pulls student identifier from top of assignment

initiate Module 3; //Begin property verification module

initiate Module 5(studentID, score); //start Module 5; send studentID and score as parameters

return(0); //terminate application after complete execution

}

**Rajah 2.6 Contoh perincian modul dan kod pseudo modul**



### 2.2.3 NDSU COMPUTER GRADING SYSTEM

Sistem penggredan berkomputer ini dilaksanakan oleh North Dakota State University, USA ([www.ndsu.edu](http://www.ndsu.edu) : 7/7/2001). Dalam sistem ini, komponen OMR (Optical Mark Reader) dilarikan oleh Information Technology Serves di IACC 140. Sistem ini hanya boleh dicapai oleh pensyarah-pensyarah NDSU dalam lingkungan kampus. Sistem ini mempunyai empat ciri penting iaitu:

- 1) Pemarkahan berkomputer bagi kertas jawapan objektif.
- 2) Program *survey* yang menghasilkan peratusan dan analisa ringkas.
- 3) *Student Rating of Instruction (SROI)*.
- 4) Bacaan data yang mudah dan senang diimbis untuk dapatkan fail teks.

Soalan objektif dijawab dalam kertas jawapan OMR yang mempunyai dua muka dan pelajar perlu menandakan jawapannya dengan menggunakan pensel 2B atau HB sahaja. Sistem diprogramkan untuk menolak pemarkahan bagi jawapan yang ditanda dengan pen. Setiap helai kertas jawapan akan diimbis oleh pengimbis OMR.

Seseorang pensyarah yang ingin menggunakan sistem ini perlu mengisikan borang INFO untuk menentukan pilihan pemarkahan, pilihan laporan, bilangan laporan yang perlu dicetak dan maklumat-maklumat lain. Sistem akan mengimbis borang ini secara automatik dan akan melakukan pemprosesan sebagaimana yang diinginkan oleh pengguna. Terdapat dua kaedah pemarkahan yang dilaksanakan iaitu pemarkahan konvensional (*conventional grading*) dan *weighted grading*:

- ♦ Pemarkahan Konvensional - Markah yang diperolehi oleh pelajar ialah jumlah soalan yang dijawab dengan betul. Setiap kertas jawapan pelajar diimbis dan dibandingkan dengan lembaran kunci (*KEY sheet*) yang ditandakan dengan jawapan betul yang disediakan oleh pensyarah.



- ◆ *Weighted Grading* - Markah diberi mengikut nilai mata atau pemberat yang diberi ke atas setiap pilihan jawapan yang disediakan (seperti yang lazim digunakan dalam kuiz-kuiz personaliti).

Sistem turut memberi pilihan untuk menanda kertas jawapan subjektif (samada soalan berstruktur atau esei). Walau bagaimanapun, pemarkahan untuk kertas ini masih secara manual. Tetapi, pensyarah boleh mengisikan terus markah yang diperolehi daripada bahagian ini dalam lembaran kunci pada medan *SPECIAL CODE* dan mengisikan markah maksimum yang boleh diperolehi bagi kertas subjektif tersebut pada ruang yang sama.

Output (laporan) daripada proses ini akan dihasilkan dalam format *tab-delimited text file* yang boleh dimport terus ke *spreadsheet* seperti Microsoft Excel. Terdapat tujuh jenis pilihan laporan iaitu:

- 1) Laporan pilihan pengguna (berdasarkan borang INFO).
- 2) Laporan maklumat pensyarah dan lembaran kunci.
- 3) Laporan taburan markah dan analisis.
- 4) Laporan taburan frekuensi mengikut soalan.
- 5) Laporan taburan frekuensi mengikut gred.
- 6) Laporan keputusan pelajar (individu).
- 7) Laporan keputusan pelajar (pilihan).

Rajah 2.7 merupakan satu contoh laporan maklumat pensyarah yang memilih pemarkahan konvensional. Rajah 2.8 pula menunjukkan contoh laporan keputusan bagi beberapa orang pelajar. Kelebihan sistem ini terlihat pada kepelbagaian format laporan yang dapat dijanakan. Ciri ini dapat memudahkan pindahan fail ke sistem pangkalan data pelajar yang lain dan memudahkan pensyarah menyimpan rekod pencapaian pelajarinya secara *soft copy*.



Summary Report for Claim Number 641  
Data Scanned on 1-14-96, Report Generated on 02APR96

```

+-----+
|               REPORT # 2               |
| Instructor Information                   |
+-----+

```

Department --> CDD

Course Number -->

Test Date --> / /

Grading Rule --> Conventional

Order of Summary Report --> by Alphabetic

Number of Copies --> 2

Number of Questions --> 50

Total Subjective Points --> 0

Ques Key Weight Ques Key Weight Ques Key Weight Ques Key Weight Ques Key Weight

1	B	1	11	A	1	21	E	1	31	A	1	41	A
2	B	1	12	A	1	22	C	1	32	B	1	42	B
3	B	1	13	B	1	23	E	1	33	A	1	43	B
4	B	1	14	E	1	24	D	1	34	A	1	44	A
5	B	1	15	A	1	25	E	1	35	A	1	45	A
6	A	1	16	A	1	26	D	1	36	A	1	46	B
7	A	1	17	A	1	27	E	1	37	A	1	47	A
8	B	1	18	D	1	28	E	1	38	A	1	48	B
9	B	1	19	A	1	29	E	1	39	A	1	49	A
10	B	1	20	E	1	30	D	1	40	A	1	50	A

**Rajah 2.7 Contoh laporan maklumat pensyarah beserta lembaran kunci**

Summary Report for Claim Number 641  
 Data Scanned on 1-14-96, Report Generated on 02APR96

```

+-----+
|               REPORT # 7               |
|               Student Summary           |
+-----+
  
```

Student Summary: CDD

Student ID-Number	Obj. Right	Obj. Wrong	Obj. Omits	Subj. Points	---- Points	Score Percent	
				Means>>===>	42.3	84.6	
3361 ELROY	39	11			39	78.0	JETSON
911	44	6			44	88.0	OPUS P
3368	38	12			38	76.0	STIMPY
6803 BARNEY	47	2	1		47	94.0	RUBBLE
** PORTION DELETED TO SAVE SPACE **							
1020 STEVE	40	10			40	80.0	DALLAS
29804 MIKE	41	9			41	82.0	BINKLEY
22305 QUEST JOHNNY		46	4			46	92.0

## Rajah 2.8 Contoh laporan keputusan pelajar (pilihan)



## 2.3 ANALISIS KESELURUHAN SISTEM

### 2.3.1 RUMUSAN DARIPADA KAJIAN SISTEM

Secara keseluruhannya, terdapat beberapa persamaan dan perbezaan yang dikongsi bersama oleh sistem-sistem yang telah dikaji sebelum ini. Jika dibandingkan antara sistem yang sedia ada, sistem yang akan dibangunkan kelak mengandungi banyak kelainan. Tetapi yang paling ketara ialah dari segi penggunaan teknologi dan bahasa pengaturcaraan di mana sistem yang akan dibangunkan hanya akan menggunakan *Visual Basics 6.0*, *Microsoft Access* dan beberapa aplikasi lain. Secara dasarnya, **SAP** adalah satu sistem baru yang kurang kompleks berbanding sistem-sistem yang telah dikaji. Maka, kaedah pembangunannya jauh lebih mudah sesuai dengan sifat penggunaannya yang lebih bersifat individu.

Ketiga-tiga sistem mempunyai konsep yang hampir sama, tetapi masih terdapat perbezaan dari segi pendekatan mengikut keperluan pengguna. Sesetengah ciri tersebut turut akan diadaptasikan oleh sistem yang bakal dibangunkan kelak.

#### ◆ Capaian sistem

Konsep capaian sistem-sistem sedia ada ini hampir sama, iaitu penggunaannya bersifat mencapah. Bagi sistem pertama yang menggunakan pakej *Mentor Ed-Soft*, pengguna-pengguna seperti ibu bapa dan pelajar boleh mencapai sistem dari kediaman masing-masing menerusi Internet. Para pendidik pula tidak menghadapi masalah untuk mencapai sistem kerana sistem telah diintegrasikan kepada semua komputer peribadi di bilik-bilik pensyarah.

Situasi yang sama bagi **CS 100-GAS**, di mana pelajar boleh mengetahui keputusan mereka melalui capaian langsung menerusi Internet dari kediaman masing-masing. Ini merupakan kelebihan bagi sistem-sistem yang dibangunkan berorientasikan *web* atau secara dalam talian.



Walau bagaimanapun, pensyarah dan tutor hanya boleh memasuki sistem dalam lingkungan fakulti sahaja. Bagi **NDSU Computer Grading System**, hanya pensyarah sahaja yang boleh memasuki sistem dan capaian hanya boleh dibuat dalam lingkungan kampus sahaja. Bagi sistem yang akan dibangunkan kelak, hanya pensyarah sahaja yang boleh mencapai sistem.

#### ◆ Sistem pangkalan data

Ketiga-tiga sistem yang dikaji menggunakan sistem pangkalan data untuk menyimpan semua data-data sistem. Sistem pangkalan data ini memudahkan mekanisme penyimpanan dan capaian kembali data-data apabila diperlukan. Sistem pangkalan yang akan dibangunkan untuk **SAP** akan menyimpan maklumat asas mengenai pelajar serta keputusan kertas ujian dan peperiksaannya. Oleh kerana seseorang pensyarah biasanya akan mengendalikan lebih daripada satu kursus, pangkalan data yang akan dibangunkan mesti berkeupayaan untuk menyimpan data yang banyak.

#### ◆ Penggunaan log masuk dan pendaftaran

Ketiga-tiga sistem ini menggunakan log masuk dan pendaftaran untuk mengawal capaian ke atas data dan penghantaran maklumat. Ciri ini juga bertujuan untuk menjaga kerahsiaan maklumat pengguna berdaftar. Bagi **SAP**, mekanisme log masuk ke dalam rangkaian tidak perlu kerana ia merupakan satu sistem individu yang tidak dihubungkan ke rangkaian LAN atau WAN setempat. Ciri ini hanya sesuai bagi sistem yang dintegrasikan ke dalam rangkaian seperti **NDSU Computer Grading System** dan **CS 100-GAS**.



Walaupun bagaimanapun, bagi **SAP**, pensyarah diminta untuk memasukkan katalaluan untuk memasuki sistem berkenaan. Walaupun **SAP** hanya berada dalam lingkungan pejabat seseorang-seseorang pensyarah, tetapi penggunaan kata laluan masih diperlukan untuk mengesahkan capaian ke dalam sistem berkenaan.

#### ♦ Antaramuka pengguna

Sistem pertama dan kedua mempunyai antaramuka yang interaktif tetapi mudah difahami; sesuai dengan sifat sistem yang berorientasikan *web*. Pakej *Mentor* dalam *Ed-Soft* memperlihatkan keseragaman dari segi stail dan rekabentuk antaramukanya. Bagi antaramuka **CS 100-GAS**, tidak banyak grafik digunakan. Penggunaan grafik yang berlebihan akan melambatkan proses memuat turun sesuatu laman web. Antaramuka input dan output bagi sistem ketiga adalah ringkas dan lebih *straightforward*. Satu prinsip umum rekabentuk antaramuka yang baik ialah antaramuka mestilah memudahkan interaksi di antara pengguna dengan sistem supaya pengguna merasa selesa. Prinsip ini akan digunakan dalam rekabentuk antaramuka bagi sistem yang akan dibangunkan kelak.

#### ♦ Ciri bantuan

Ciri ini terdapat di dalam semua sistem yang dikaji. Sistem ini amat penting untuk memberikan panduan kepada pengguna untuk menggunakan sistem. Ciri ini akan diterapkan di dalam pembangunan sistem yang akan dibangunkan. Walaupun sistem baru akan dilengkapi manual pengguna dan antaramuka yang ringkas, ciri bantuan akan menambahkan lagi kebolegunaan sistem.

## ◆ Ciri atas talian (*online*)

Ciri ini banyak digunakan di dalam sistem aplikasi berasaskan web pada masa kini. Malah, dua sistem pertama yang dikaji juga mempunyai ciri ini. Kebaikan ciri ini ialah memudahkan pengemaskinian dan capaian terhadap data. Kelemahannya, jika berlaku kegagalan dalam rangkaian, sistem tidak boleh berfungsi.

Sebagai sistem yang lebih bersifat individu, **SAP** tidak akan menggunakan ciri ini sebagai konsep utama pembangunannya. Bersifat *stand-alone*, seseorang pensyarah bertanggungjawab sepenuhnya dalam menguruskan sistem itu sendiri seperti dapat mengemaskini data pelajarinya dengan mudah serta dapat mengurangkan risiko pencerobohan dan pengubahsuaian data pelajar secara tidak sah daripada pihak yang tidak bertanggungjawab.

Selain daripada ciri-ciri berkenaan, **SAP** akan memiliki ciri carian yang biasa terdapat dalam kebanyakan sistem untuk mempercepatkan proses pencapaian data yang disimpan di dalam pangkalan data. Oleh kerana data yang tersimpan dalam pangkalan data adalah banyak, kaedah pencarian yang baik akan menyenangkan pengguna membuat capaian ke atas data tertentu tanpa perlu melayari kesemua data yang ada. Ciri ini akan diberi perhatian dalam pembangunan **SAP**.



## 2.4 KUALITI-KUALITI YANG CUBA DIPENUHI OLEH **SAP**

Berikut adalah kualiti-kualiti yang perlu diambil perhatian dalam pembangunan **SAP** untuk menghasilkan satu sistem penggredan dan penilaian yang praktikal dan memenuhi objektif pembangunannya:

KUALITI	PENERANGAN
Kebolehfahaman ( <i>understandibility</i> )	Sejauh manakah proses penggredan dan analisis yang berlaku dalam sistem ditakrifkan secara tepat dan adakah ia mudah difahami umum?
Kebolehgantungan ( <i>reliability</i> )	Adakah proses direkabentuk untuk membolehkan sebarang ralat dikenalpasti dan dielakkan daripada menghasilkan output gred yang tidak tepat?
Kepantasan ( <i>rapidity</i> )	Sejauh mana pantasny sistem akan menjanakan gred, menyimpan rekod gred dan menjanakan graf untuk dianalisis?
Kebolegunaan ( <i>usability</i> )	Adakah sistem yang dibangunkan ini bersifat praktikal dan boleh digunakan oleh mana-mana pensyarah sekalipun?
Kebolehnampakkan ( <i>visibility</i> )	Adakah aktiviti dalaman sistem dapat menghasilkan output yang boleh nampak untuk mengesahkan samada proses itu benar-benar berjaya?
Kebolehsenggaraan ( <i>maintainability</i> )	Adakah sistem/proses yang direka boleh berevolusi mengikut keperluan pengguna?
Kecekapan ( <i>efficiency</i> )	Adakah sistem dapat memenuhi kesemua objektif pembangunannya?
Kekebalan ( <i>robustness</i> )	Adakah keutuhan data terjamin walaupun sistem menghadapi masalah?

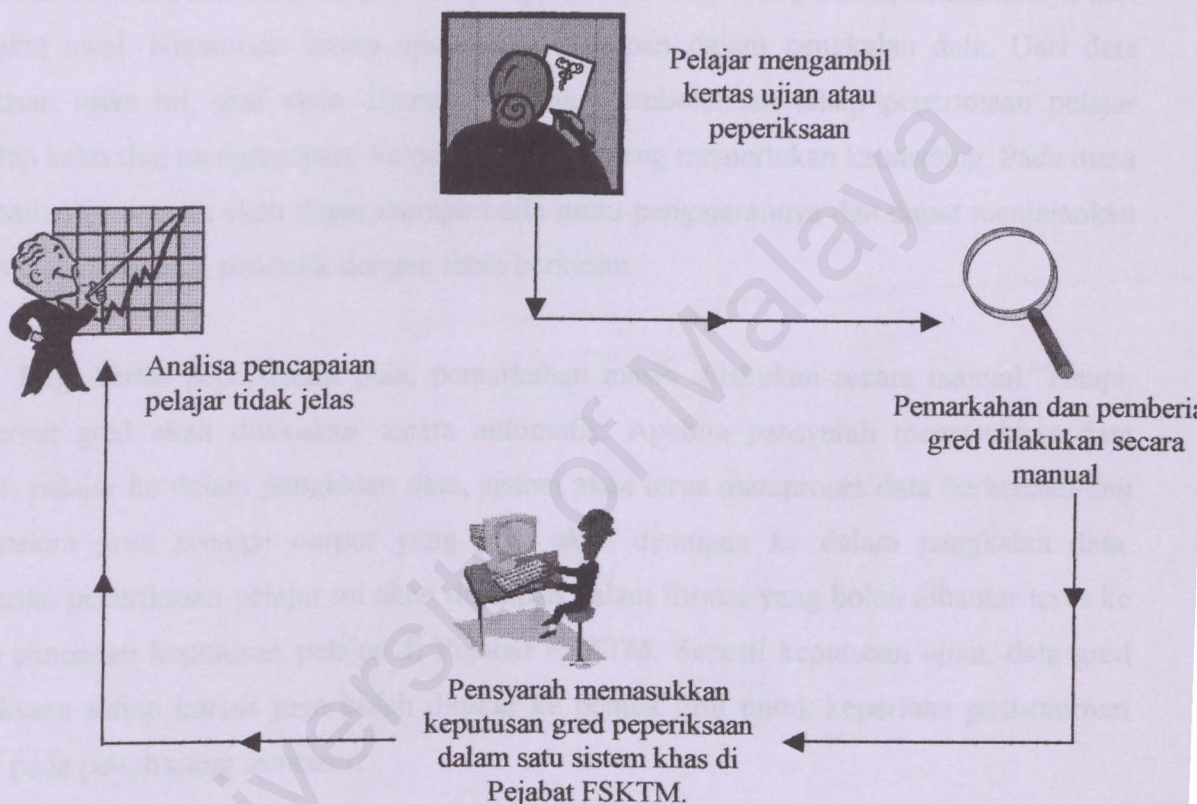
**Jadual 2.1 : Kualiti-kualiti yang cuba dicapai oleh **SAP****



## 2.5 GAMBARAN PROSES **SAP** vs SISTEM SEMASA

Berikut adalah gambaran proses semasa yang diimplementasikan oleh para pensyarah dalam kerja-kerja penggredan dan simpanan data gred pelajar:

**Rajah 2.9** Gambaran proses semasa



Daripada gambaran ini dapat dilihat bahawa prosedur pemberian gred dan analisa pencapaian adalah tidak jelas. Malah, tiada pendekatan yang sistematik untuk memantau prestasi pelajar daripada awal kursus sehingga peperiksaan. Sekiranya keputusan ujian pelajar turut diberikan perhatian, pensyarah dapat menganalisis pencapaian dan pemahaman para pelajar terhadap kelas kendaliannya.



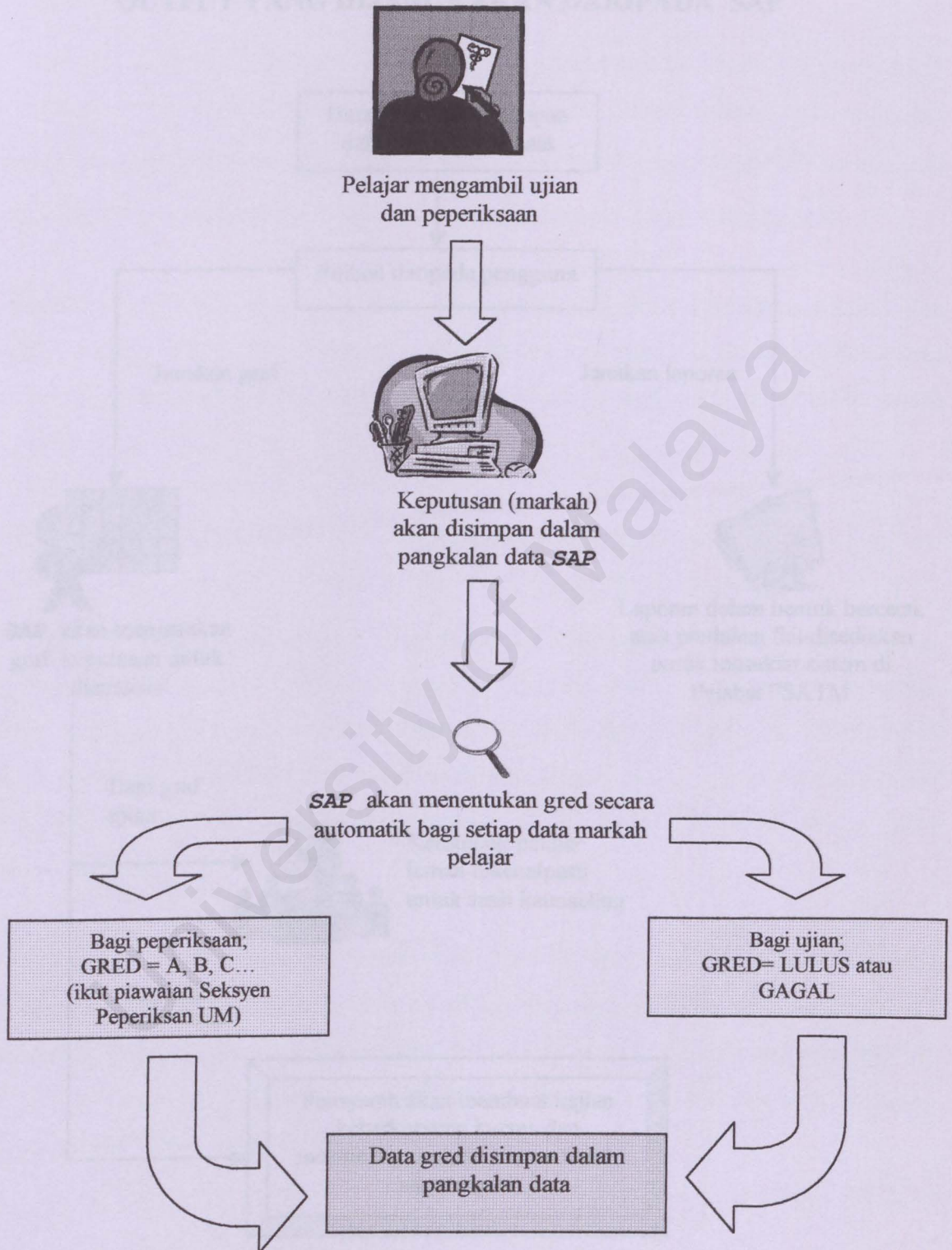
Para pensyarah juga menggunakan masa yang banyak untuk menanda kertas peperiksaan dan memberikan gred secara manual. Mereka juga terpaksa pergi sendiri ke Pejabat FSKTM untuk memasukkan keputusan peperiksaan pelajar bagi setiap kursus kendaliannya kerana satu-satunya sistem tersebut berada di situ.

Kelemahan-kelemahan inilah yang cuba diatasi oleh **SAP**. Melalui sistem ini, pensyarah akan dapat memantau prestasi pelajar-pelajar bagi setiap kursus kendaliannya dari peringkat awal. Keputusan setiap ujian akan disimpan dalam pangkalan data. Dari data keputusan ujian ini, graf akan dijanakan untuk membolehkan tahap penerimaan pelajar terhadap kelas dan mengenalpasti kelompok pelajar yang memerlukan kaunseling. Pada masa yang sama, pensyarah akan dapat memperbaiki mutu pengajarannya dan dapat menjalankan peranannya sebagai pendidik dengan lebih berkesan.

Bagi kertas peperiksaan pula, pemarkahan masih dilakukan secara manual. Tetapi, pemberian gred akan dilakukan secara automatik. Apabila pensyarah memasukkan data markah pelajar ke dalam pangkalan data, sistem akan terus memproses data berkenaan dan menjanakan gred sebagai output yang juga akan disimpan ke dalam pangkalan data. Keputusan peperiksaan pelajar ini akan disimpan dalam format yang boleh dihantar terus ke sistem simpanan keputusan pelajar di Pejabat FSKTM. Seperti keputusan ujian, data gred peperiksaan setiap kursus juga boleh ditukar ke bentuk graf untuk keperluan post-mortem kursus pada penghujung semester.

Berikut pula adalah gambaran proses bagi **SAP**. Sistem yang akan dibangunkan ini adalah lebih sistematik dan mempunyai pertambahan beberapa fungsi untuk meningkatkan keupayaan penggunaanya.

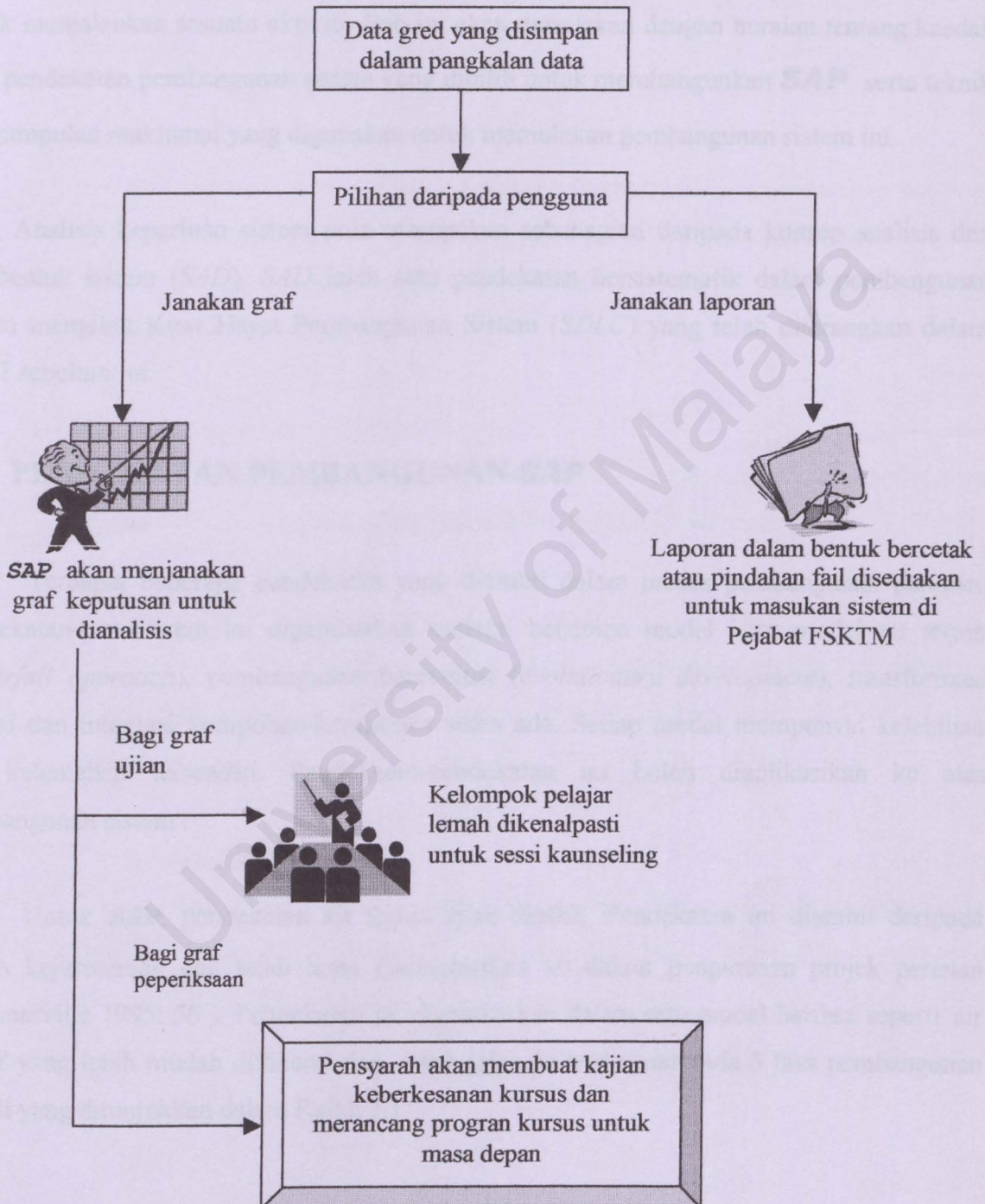
Rajah 2.10 : Gambaran proses **SAP**





Rajah 2.11

OUTPUT YANG DIJANGKAKAN DARIPADA SAP



## BAB 3 : METODOLOGI DAN ANALISIS KEPERLUAN SISTEM

Metodologi didefinisikan sebagai kaedah atau pendekatan bersistematik yang dipilih untuk menjalankan sesuatu aktiviti. Bab ini akan dimulakan dengan huraian tentang kaedah atau pendekatan pembangunan sistem yang dipilih untuk membangunkan **SAP** serta teknik pengumpulan maklumat yang digunakan untuk memulakan pembangunan sistem ini.

Analisis keperluan sistem pula merupakan sebahagian daripada konsep analisis dan rekabentuk sistem (*SAD*). *SAD* ialah satu pendekatan bersistematik dalam pembangunan sistem mengikut Kitar Hayat Pembangunan Sistem (*SDLC*) yang telah diterangkan dalam Bab 1 sebelum ini.

### 3.1 PENDEKATAN PEMBANGUNAN **SAP**

Terdapat beberapa pendekatan yang diambil dalam proses pembangunan perisian. Pendekatan-pendekatan ini digambarkan melalui beberapa model iaitu model air terjun (*waterfall approach*), pembangunan berevolusi (*evolutionary development*), transformasi formal dan integrasi komponen-komponen sedia ada. Setiap model mempunyai kelebihan dan kelemahan tersendiri. Pendekatan-pendekatan ini boleh diaplikasikan ke atas pembangunan sistem .

Untuk **SAP**, pendekatan air terjun telah dipilih. Pendekatan ini diambil daripada proses kejuruteraan dan telah lama diadaptasikan ke dalam pengurusan projek perisian (Sommerville 1995: 56 ). Pendekatan ini digambarkan dalam satu model berfasa seperti 'air terjun' yang lebih mudah difahami dan lebih jelas. Ia terdiri daripada 5 fasa pembangunan seperti yang ditunjukkan dalam Rajah 3.1.



1) *Analisis dan definisi keperluan*

Perkhidmatan sistem, kekangan dan matlamat sistem ditetapkan dengan bantuan bakal pengguna. Ia dipersembahkan dalam bentuk yang mudah difahami oleh kedua-dua pembangun dan pengguna.

2) *Rekabentuk sistem dan perisian*

Proses rekabentuk sistem dibahagikan kepada keperluan perkakasan dan perisian. Ini menyediakan asas bagi senibina keseluruhan sistem. Rekabentuk perisian bertanggungjawab mempersembahkan fungsi perisian perisian dalam bentuk program yang boleh dilarikan.

3) *Implementasi dan pengujian unit*

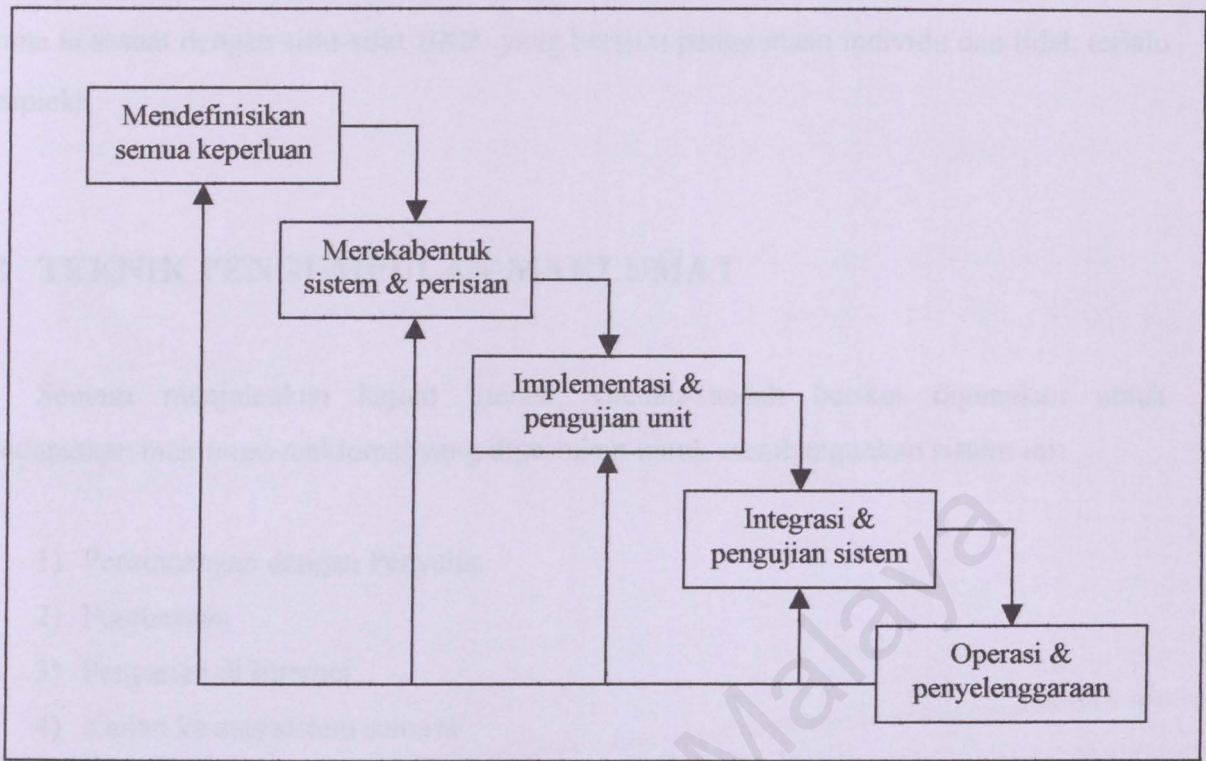
Perisian dibangunkan sebagai unit-unit dalam satu set program dan setiap unit diuji untuk memastikan setiap satu mencapai spesifikasinya.

4) *Integrasi dan pengujian sistem*

Unit-unit program diintegrasikan dan diuji sebagai satu sistem yang lengkap untuk memastikan keperluan sistem dipenuhi. Sistem akan dihantar kepada pengguna.

5) *Operasi dan penyelenggaraan*

Secara umumnya, sistem dipasang dan digunakan. Penyelenggaraan dilakukan apabila sebarang ralat dikenalpasti dan mengubahsuai perincian mengikut keperluan pengguna.



**Rajah 3.1 Model air terjun**

Secara praktikalnya, fasa-fasa ini saling bertindih dan memerlukan maklumat daripada satu sama lain. Selain itu, mungkin juga berlaku ulangan yang berulang kali kerana apabila berlaku kesilapan pada satu-satu fasa, fasa (atau fasa-fasa) sebelumnya terpaksa dirujuk semula. Contohnya, jika sesuatu ralat operasi dikenalpasti semasa fasa operasi, mungkin fasa rekabentuk akan dirujuk semula untuk mengesan bahagian yang mengandungi kelemahan. Ini mungkin satu kelebihan bagi sistem yang dibangunkan menggunakan pendekatan ini. Walau bagaimanapun, jika ulangan fasa diulangi terlalu kerap, ia membayangkan struktur rekabentuk sistem yang lemah serta menyukarkan penentuan titik rujukan (*checkpoint*) untuk penyediaan laporan dan perancangan.



Walau bagaimanapun, pendekatan ini dipilih sebagai model pembangunan **SAP** kerana ia sesuai dengan sifat-sifat **SAP** yang bersifat penggunaan individu dan tidak terlalu kompleks.

## **3.2 TEKNIK PENGUMPULAN MAKLUMAT**

Semasa menjalankan kajian literasi, kaedah-kaedah berikut digunakan untuk mendapatkan maklumat-maklumat yang diperlukan untuk membangunkan sistem ini:

- 1) Perbincangan dengan Penyelia.
- 2) Pembacaan
- 3) Pencarian di Internet.
- 4) Kajian ke atas sistem semasa

### **3.2.1 Perbincangan dengan Penyelia**

Penyelia merupakan sumber maklumat primer bagi projek ini kerana sistem yang dibangunkan perlu menepati spesifikasi dan kehendak beliau. Melalui beberapa sesi perbincangan dengan Cik Norisma Idris, gambaran umum tentang sistem (yang asalnya dinamakan *Sistem Analisis Penggredan*) diberikan serta maklumat-maklumat asas tentang sistem yang akan dibangunkan kelak.

### **3.2.2 Pembacaan**

Pembacaan dan kajian daripada hasil bacaan dibuat ke atas dokumen-dokumen yang berkaitan dengan projek ini iaitu '*penggredan dan analisis*'. Material bacaan terdiri daripada buku-buku, jurnal-jurnal dan tesis-tesis yang terdapat di Perpustakaan Utama, Universiti Malaya dan di bilik dokumen Fakulti Sains Komputer dan Teknologi Maklumat.

Selain daripada itu, pembacaan yang dibuat ini juga ada berdasarkan kepada sumber - sumber dan bahan - bahan rujukan simpanan sendiri seperti buku - buku teks dan sebagainya lagi. Beberapa tesis yang dihasilkan oleh pelajar-pelajar sebelum ini turut menjadi rujukan

utama untuk pembangunan sistem ini untuk mendapatkan stail laporan, perincian maklumat dan sebagainya. Maklumat-maklumat yang menarik dan penting dikaji dan direkodkan sebagai sumber maklumat.

### 3.2.3 Kajian di Internet

Melalui carian di Internet, tiga sistem yang hampir menyamai konsep sistem yang akan dibangunkan diperolehi. Carian di Internet adalah antara kaedah yang agak efektif dalam mendapatkan sumber maklumat kerana Internet merupakan tapak pencarian maklumat yang mudah dan cepat. Malah, dua daripada sistem yang menjadi bahan kajian literasi ( rujuk Bab 2 ) diperolehi daripada Internet; iaitu laman web yang dibangunkan oleh pembangun-pembangun sistem ( samada para pelajar atau organisasi) untuk berkongsi pengalaman mereka kepada pelayar-pelayar Internet. Kajian yang dibuat ke atas sistem-sistem yang diperolehi melalui Internet ini menghasilkan banyak penemuan-penemuan yang penting untuk diambilkira dalam pembangunan **SAP**.

### 3.2.4 Kajian ke atas sistem semasa

Di FSKTM, belum ada sistem seumpama **SAP** dibangunkan untuk kegunaan pensyarah. Walau bagaimanapun, prosedur biasa yang diamalkan oleh pensyarah dalam penetapan gred dan analisa pencapaian pelajar dikaji kerana ia menghasilkan gambaran penting bagi fungsi sistem yang akan dibangunkan. Maklumat tentang 'sistem' semasa ini diperolehi melalui perbincangan dengan Penyelia dan beberapa pensyarah lain. Situasi semasa dikaji bagi mengenalpasti kelemahan yang wujud dan cuba mencadangkan ciri-ciri pembaharuan bagi sistem yang dicadangkan.



### 3.3 ANALISIS KEPERLUAN

Analisis keperluan dibahagikan kepada 2 bahagian iaitu keperluan fungsian dan keperluan bukan fungsian. Keperluan fungsian merujuk kepada bagaimana sistem itu akan beroperasi manakala keperluan bukan fungsian merujuk kepada perkara-perkara lain yang cuba dipenuhi oleh sistem.

#### 3.3.1 KEPERLUAN FUNGSIAN *SAP*

*SAP* dibahagikan kepada 4 modul iaitu modul *Kursus*, modul *Pelajar*, modul *KepUjian* dan modul *KepPep*. Keempat-empat modul ini diwujudkan selari dengan struktur pangkalan data di mana setiap modul akan merujuk kepada fail *Kursus*, fail *Pelajar*, fail *KepUjian* dan fail *KepPep* masing-masing.

Untuk mengesahkan capaian ke dalam *SAP*, pengguna perlu memasukkan nama pengguna dan katalaluan. Pada kali pertama menggunakan sistem ini, pengguna akan diminta mendaftarkan nama pengguna (biasanya nama pertama pengguna) dan katalaluan. Secara logiknya, setiap pensyarah akan mempunyai katalaluannya. Maka, hanya beliau sahaja yang boleh melihat maklumat pelajar dan maklumat keputusan para pelajar yang mengambil kertas beliau.

Penggunaan log-in seperti ini hampir menyamai konsep log-in ke atas antaramuka Windows pada komputer peribadi milik individu. Hanya apabila nama pengguna dan katalaluan yang tepat dimasukkan, barulah capaian ke atas data-data dalam sistem dibenarkan.

### 3.3.1.1 Modul *Kursus*

Perkara pertama yang perlu dilakukan oleh pengguna untuk mencapai data-dat data dalm rekod sistem ialah memasukkan kod kursus. Input pengguna akan dirujuk kepada fail *Kursus* yang mengandungi maklumat kod dan nama kursus. Jika kod kursus yang diinputkan wujud dalam rekod, capaian ke tahap seterusnya akan diteruskan; iaitu capaian ke atas rekod pelajar, rekod keputusan ujian dan rekod keputusan peperiksaan.

Seterusnya, jika kod kursus yang diinputkan tiada dalam rekod, mesej ralat akan dipaparkan. Jika pengguna ingin menambah atau menghapuskan mana-mana kod kursus yang ada dalam simpanan, sistem akan memandu pengguna dan menjalankan operasi tersebut. Apabila satu-satu rekod kod ini dihapuskan, rekod pelajar, rekod keputusan ujian dan peperiksaan yang berkaitan dengan rekod terbabit dihapuskan secara automatik.

### 3.3.1.2 Modul *Pelajar*

Selepas melepasi modul *Kursus* tadi, pengguna diberikan pilihan untuk mencapai rekod pelajar atau terus mencapai rekod keputusan pelajar. Pemilihan rekod pelajar akan membenarkan pengguna melihat senarai nama pelajar beserta nombor matrik. Pengguna dibenarkan menambah rekod nama pelajar baru atau menghapuskan nama pelajar yang sudah menarik diri daripada kursus. Operasi ini dijalankan ke atas fail *Pelajar* dalam pangkalan data. Jika pengguna tidak berminat untuk mengemaskinikan rekod pelajar, pengguna boleh terus memilih untuk mencapai rekod keputusan ujian atau keputusan peperiksaan pelajar.



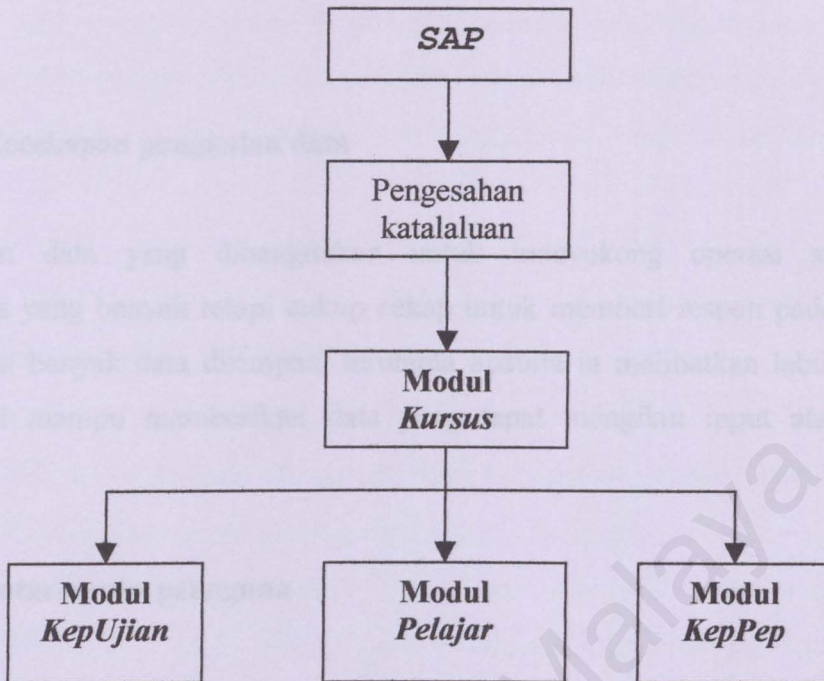
### 3.3.1.3 Modul *KepUjian*

Jika pengguna memilih untuk mencapai rekod keputusan ujian pelajar, beliau dibenarkan untuk mengemaskinikan rekod atau mendapatkan laporan daripada rekod sedia ada dalam bentuk graf atau laporan bertulis. Untuk mengemaskini, pengguna dibenarkan memasukkan data markah ke dalam rekod keputusan. Apabila pengguna mengeluarkan arahan untuk menjanakan gred, sistem akan menjanakan gred dalam bentuk penyataan lulus atau gagal. Graf akan dijanakan daripada data lulus-gagal ini.

### 3.3.1.4 Modul *KepPep*

Jika pengguna memilih untuk mencapai rekod keputusan peperiksaan pelajar, beliau dibenarkan untuk mengemaskinikan rekod atau mendapatkan laporan daripada rekod sedia ada dalam bentuk graf atau laporan bertulis. Untuk mengemaskini, pengguna dibenarkan memasukkan data jumlah markah peperiksaan akhir ke dalam rekod keputusan. Apabila pengguna mengeluarkan arahan untuk menjanakan gred, gred akan dijanakan secara automatik bagi setiap markah akhir. Graf akan dijanakan daripada data gred.

Berikut adalah gambarajah yang menunjukkan bagaimana keempat-empat modul berkenaan membentuk **SAP**:



**Rajah 3.2** Gambarajah pemodulan **SAP**

### **3.3.2 KEPERLUAN BUKAN FUNGSIAN *SAP***



Selain daripada operasi-operasi asas yang diterangkan dalam keperluan fungsian tadi, **SAP** turut memberikan perhatian kepada beberapa ciri lain yang mempengaruhi kecekapan sistem.

### 3.3.2.1 Kecekapan pangkalan data

Pangkalan data yang dibangunkan untuk menyokong operasi sistem dapat menampung data yang banyak tetapi cukup cekap untuk memberi respon pada kadar yang segera. Meskipun banyak data disimpan, terutama apabila ia melibatkan lebih daripada 3 kursus, ia mesti mampu memberikan data yang tepat mengikut input atau keperluan pengguna.

### 3.3.2.2 Antaramuka pengguna

Antaramuka yang dibangunkan akan cuba memudahkan interaksi pengguna dengan arahan-arahan yang mudah difahami pengguna. Ia perlu praktikal sistem ini adalah untuk mempercepatkan tugas penyimpanan dan penentuan gred pelajar. Antaramuka **SAP** menggunakan ikon dan latar-belakang berwarna yang menambahkan nilai estetika antaramuka berkenaan.

### 3.3.2.3 Ketepatan

Oleh kerana modul *KepUjian* dan *KepPep* akan menetapkan gred ke atas markah akhir pelajar, algoritma pengaturcaraan yang efektif direka untuk memastikan gred yang dijanakan mengikut piawaian daripada Seksyen Peperiksaan UM. Oleh kerana gred dijanakan secara automatik, mekanisme penetapan itu mesti jelas dan tepat.

### 3.3.2.3 Output yang efektif

Selain daripada gred yang dijanakan secara automatik, graf juga merupakan output akhir yang penting daripada sistem ini. Dua graf akan dijanakan iaitu graf ujian berdasarkan lulus-gagal dan graf keputusan peperiksaan berdasarkan gred. Penjana graf ini juga memerlukan algoritma pengaturcaraan yang efektif untuk membolehkan graf dipaparkan di skrin. Graf juga mesti boleh disimpan dalam pangkalan data untuk dicetak apabila diperlukan.

### 3.4 ANALISA KEPERLUAN **SAP**

Oleh kerana **SAP** merupakan satu sistem yang bersifat *stand-alone*, aplikasi peringkat tinggi seperti ASP (*Active Server Pages*) dan pakej-pakej berorientasikan web yang lain tidak diberikan keutamaan. Begitu juga bahasa pengaturcaraan seperti Java mahupun C++ kerana sistem yang ringkas seperti **SAP** memerlukan antaramuka berasaskan Windows kerana majoriti pengguna komputer biasa menggunakan antaramuka sedemikian.

Oleh itu, **SAP** akan dibangunkan menggunakan Visual Basic 6.0 untuk pembangunan antaramuka dan Microsoft Access 2000 sebagai pangkalan data. Selain itu, beberapa aplikasi lain yang dirasakan sesuai dan diperlukan akan turut digunakan. Walau bagaimanapun, kedua-dua aplikasi ini akan diberikan tumpuan.

#### 3.4.1 VISUAL BASIC 6.0

Visual Basic (VB) adalah kembangan daripada bahasa pengaturcaraan BASIC yang dibangunkan oleh Microsoft Corporation pada tahun 1991. Ia dibangunkan selaras dengan



trend baru yang melanda dunia perkomputeran pada masa itu iaitu penggunaan GUI pada penghujung 1980-an (Deitel, Deitel dan Nieto 1999: 7).

VB adalah bahasa pengaturcaraan yang menyediakan ciri-ciri seperti GUI, *event handling*, capaian ke Win32 API, ciri berorientasikan objek, pengendalian ralat, pengaturcaraan berstruktur dan sebagainya (Deitel, Deitel dan Nieto 1999: 9). Program-program yang dibangunkan menggunakan VB direka dalam IDE (*Integrated Development Environment*) di mana proses mereka, melarikan dan *debug* dapat dilakukan serentak. Ini memberikan kelebihan dalam pembangunan **SAP** kerana pembangunan antaramuka dan pengaturcaraan dapat dilakukan serentak.

VB menggunakan konsep pengaturcaraan visual di mana pengaturcara berpeluang merekabentuk GUI hanya dengan mengklik dan memilih menggunakan tetikus. Dengan ini, masa untuk menulis aturcara untuk menjanakan objek seperti menu, butang borang, pemilihan warna dan sebagainya dapat dijimatkan. Selain itu, antaramuka pengguna dapat dibangunkan dengan lebih menarik dan lebih kreatif. Pengaturcara tidak perlu menjadi seorang pakar pengaturcara Windows untuk merekabentuk antaramuka seperti Windows.

Aturcara VB juga lebih ringkas dan mudah. Kod yang ringkas diperlukan untuk menggerakkan *event* yang diinginkan oleh pengaturcara. Sebagai contoh, dalam satu aturcara untuk menukar warna 'lampu' (yang diwakili oleh satu butang) apabila butang tersebut diklik oleh pengguna, kod VBnya adalah seperti berikut:

```
Private Sub Pushbutton_Click()
```

```
WarningLight.FillColor = &HC000&  
Pushbutton.Caption = "Turn & Off"
```

```
End Sub
```

Keistimewaan pengaturcaraan dalam VB ialah ia bersifat *event-driven* di mana pengguna yang menentukan bagaimana sesuatu program akan berjalan. Contohnya, dalam aplikasi pelayar web (*web browser*), pelayar akan memaparkan satu-satu muka web *by default* dan akan kekal statik sehingga pengguna menentukan tindakan yang seterusnya dengan mengklik mana-mana butang yang sesuai. Dalam kes **SAP**, pengguna akan menentukan apa yang harus dibuat seterusnya.

Secara keseluruhannya, VB amat sesuai digunakan dalam pembangunan sistem ini kerana VB menggabungkan pembangunan antaramuka dan pengaturcaraan pada satu masa. VB menyenangkan rekabentuk antaramuka bergrafik dan bahasa pengaturcaraannya yang lebih ringkas berbanding C++ dan Java membolehkan output bergrafik (graf) dijanakan dengan lebih mudah. Versi yang digunakan ialah versi 6.0, iaitu versi terkini VB yang dibangunkan pada 1998. Secara kebetulan, aplikasi VB 6.0 pernah digunakan semasa projek Latihan Industri dahulu. Maka, diharapkan fasa pengkodan sistem kelak dapat dilaksanakan dengan lebih baik.

### 3.4.2 Microsoft Access 2000

Seperti VB 6.0, Microsoft Access 2000 juga pernah digunakan semasa projek Latihan Industri dahulu. Tambahan pula, ia merupakan sebahagian daripada pakej aplikasi Microsoft



Office, maka ia lebih berkeupayaan untuk 'bekerjasama' dengan pakej Microsoft yang lain (dalam kes ini, Visual Basic).

Ia adalah sistem pengurusan pangkalan data berhubung (RDBMS) yang dibina oleh Microsoft untuk syarikat kecil atau pengguna biasa. Perisian ini digunakan untuk menyimpan data dalam format berhubung. Dengan paradigma antaramuka capaian data seperti Remote Data Object (RDO) dan Data Access Object (DAO), Microsoft Access boleh digunakan sebagai pangkalan data di dalam senibina pelanggan/pelayan atau senibina sistem *n-tier*. Ciri-ciri ini menyediakan antaramuka yang baik yang dibangunkan dengan jadual dan hubungan.

Microsoft Access juga mempunyai ciri dan fungsi untuk menerbitkan aplikasi pangkalan data di dalam Intranet dan Internet (Kroenke 1998: 411). Ia juga mempunyai semua komponen yang diperlukan untuk mentadbir dan mengendalikan sejumlah besar data dengan mudah (Apps 1996: 5).

### **3.5 KEPERLUAN MASA PEMBANGUNAN**

#### **3.5.1 Perkakasan**

Konfigurasi yang dicadangkan ialah:

- Komputer peribadi dengan pemproses Intel Pentium III; sekurang-kurangnya 450 MHz.
- 32 MB RAM.
- Peranti-peranti asas seperti papan kekunci, tetikus, pemacu cakera keras dan pemacu cakera liut 3 1/2.
- CD Writer.

### 3.5.2 Perisian

- Windows 98 atau yang lebih tinggi.
- Visual Basic 6.0 Edisi Profesional
- Microsoft Access 2000
- Perisian-perisian lain yang dapat menyokong larian sistem.

## 3.6 KEPERLUAN MASA LARIAN

### 3.6.1 Perkakasan

Konfigurasi yang dicadangkan ialah:



- Mikrokomputer dengan pemrosesor Intel Pentium III 450MHz.
- Sekurang-kurangnya 32MB RAM.
- Storan cakera keras sekurang-kurangnya 800MB.
- Pemacu cakera keras dan pemacu cakera liut 3 1/2.
- Pencetak

### 3.6.2 Perisian

- Windows 98 atau yang lebih tinggi.
- Visual Basic 6.0
- Microsoft Access 2000
- Perisian-perisian lain yang dapat menyokong larian sistem seperti pakej Office itu sendiri.

## BAB 4 : REKABENTUK SISTEM

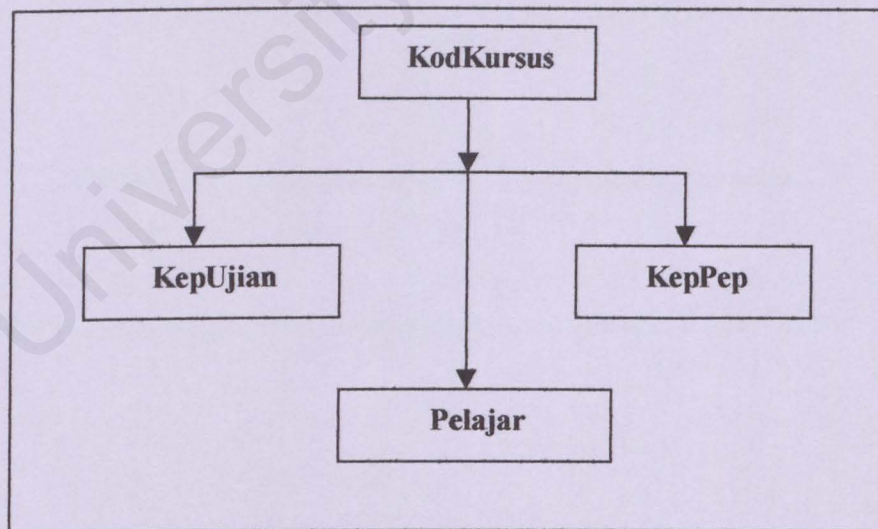
Rekabentuk adalah satu proses yang memerlukan pemikiran kreatif disertai pemahaman dan kebolehan semulajadi perekabentuk bagi menukarkan masalah kepada sesuatu bentuk penyelesaian. Kemahiran rekabentuk boleh dipelajari melalui pengalaman dan mengkaji sistem yang sedia ada.

### 4.1 REKABENTUK PROSES

Rekabentuk proses adalah berdasarkan kepada rekabentuk aliran data berorientasikan rekabentuk. Rekabentuk ini juga dikenali sebagai rekabentuk berstruktur.

#### 4.1.1 CARTA STRUKTUR PROSES

Carta struktur menunjukkan pengabstrakan peringkat tinggi di dalam spesifikasi sistem. Carta ini digunakan untuk menerangkan komponen-komponen yang terdapat di dalam sistem serta jenis capaian yang boleh dilakukan **SAP**:



Rajah 4.1 Carta struktur **SAP**



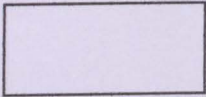
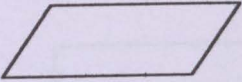
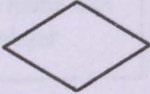
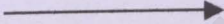
Rajah di atas menggambarkan bagaimana **SAP** mula beroperasi setelah pengesahan nama pengguna dan katalaluan diperolehi. Rajah ini menunjukkan sistem ini dibahagikan kepada 4 modul yang dibahagikan mengikut struktur pangkalan data sistem yang dinamakan **PD-SAP**. Sila rujuk bahagian 4.3 Struktur Pangkalan Data **PD-SAP** untuk mendapatkan huraian bagi mendapatkan perincian struktur dan setiap fail.

Sistem dibahagikan kepada 4 tahap capaian (rujuk rajah 4.1). Pengguna perlu membuat pilihan kursus dengan menginputkan *KodKursus* dan sistem akan memproses permintaan pengguna untuk mencapai tahap sistem seterusnya. Kemudian, pengguna diberi pilihan untuk memilih capaian ke atas rekod maklumat pelajar bagi kursus terbabit ataupun rekod keputusan pelajar.

Bagi rekod keputusan pelajar, pengguna boleh memilih samada *KepUjian* untuk keputusan ujian atau *KepPep* untuk keputusan peperiksaan. Walau bagaimanapun, pengguna boleh juga mencapai terus rekod ini selepas daripada memilih *KodKursus* tadi.

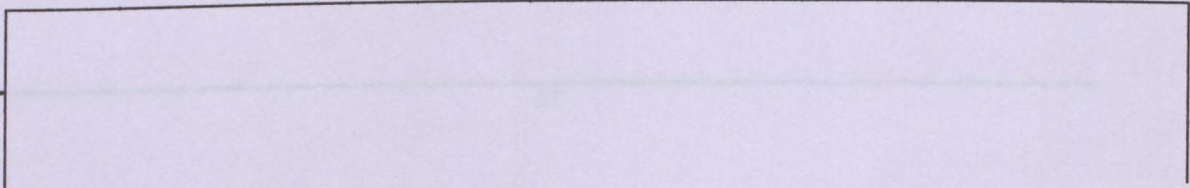
4.1.2 CARTA ALIR PROSES

Carta alir memberikan gambaran cara aliran bagi setiap modul di dalam sistem secara am. Berikut adalah notasi yang biasa digunakan dalam merekabentuk carta alir proses:

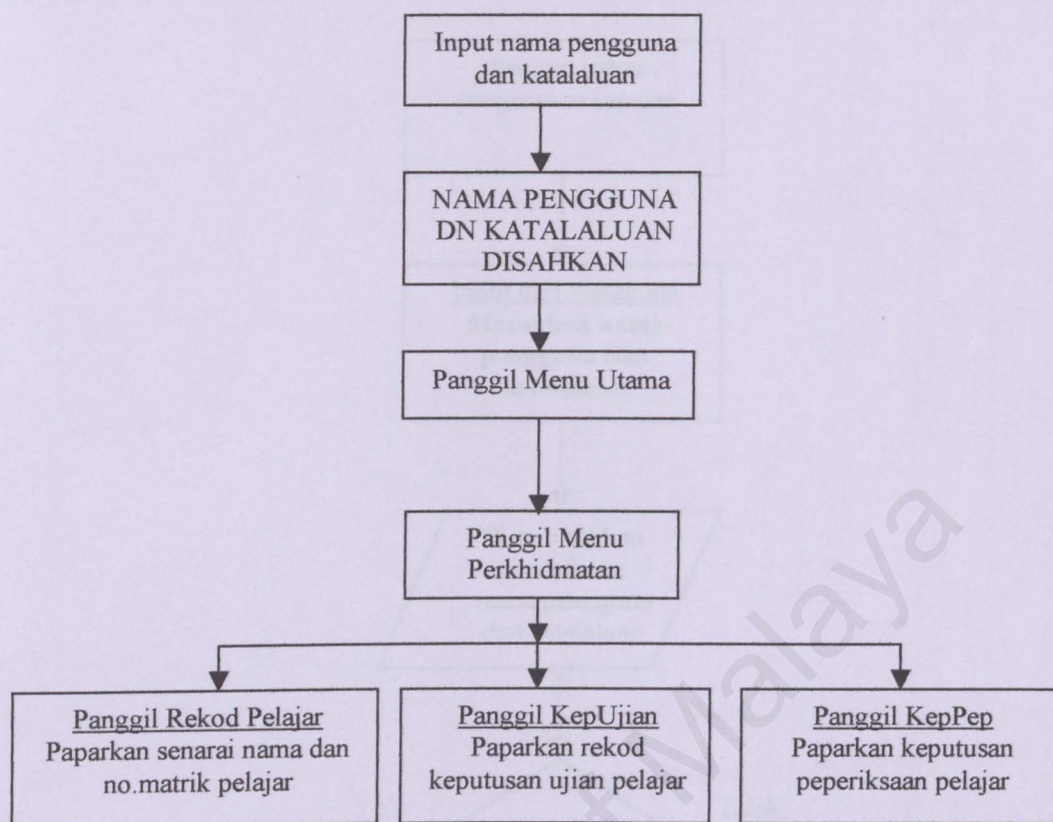
Simbol	Penerangan
	Proses yang melaksanakan operasi di dalam <i>SAP</i>
	Entiti dimana kejadian berlaku
	Pilihan perlaksanaan
	Mewakili proses perlaksanaan atau pemilihan modul atau menu yang seterusnya.

Jadual 4.1 : Simbol dalam carta alir dan penerangannya.

Rajah-rajab seterusnya menunjukkan bagaimana aliran proses berlaku dalam *SAP*.

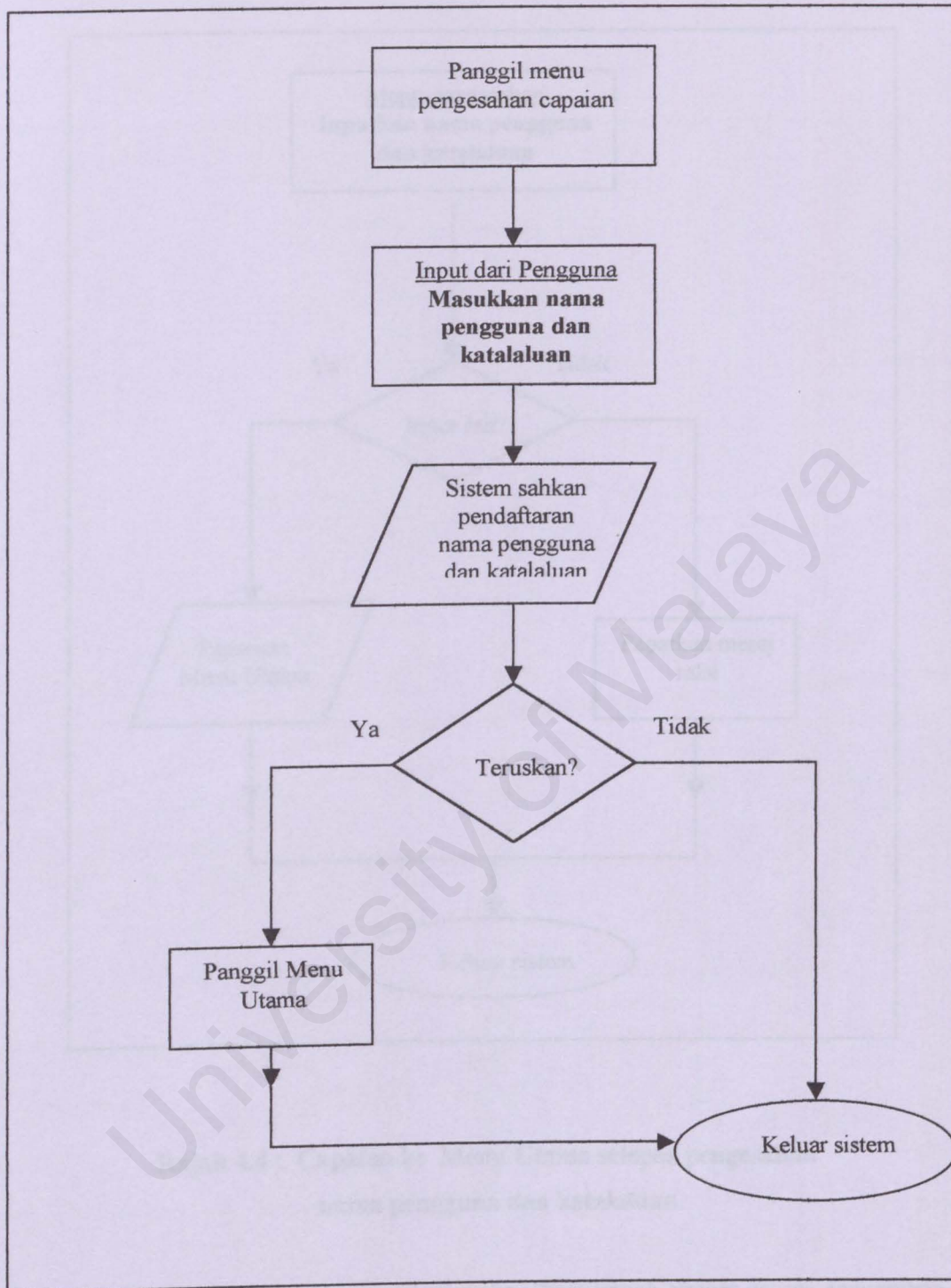






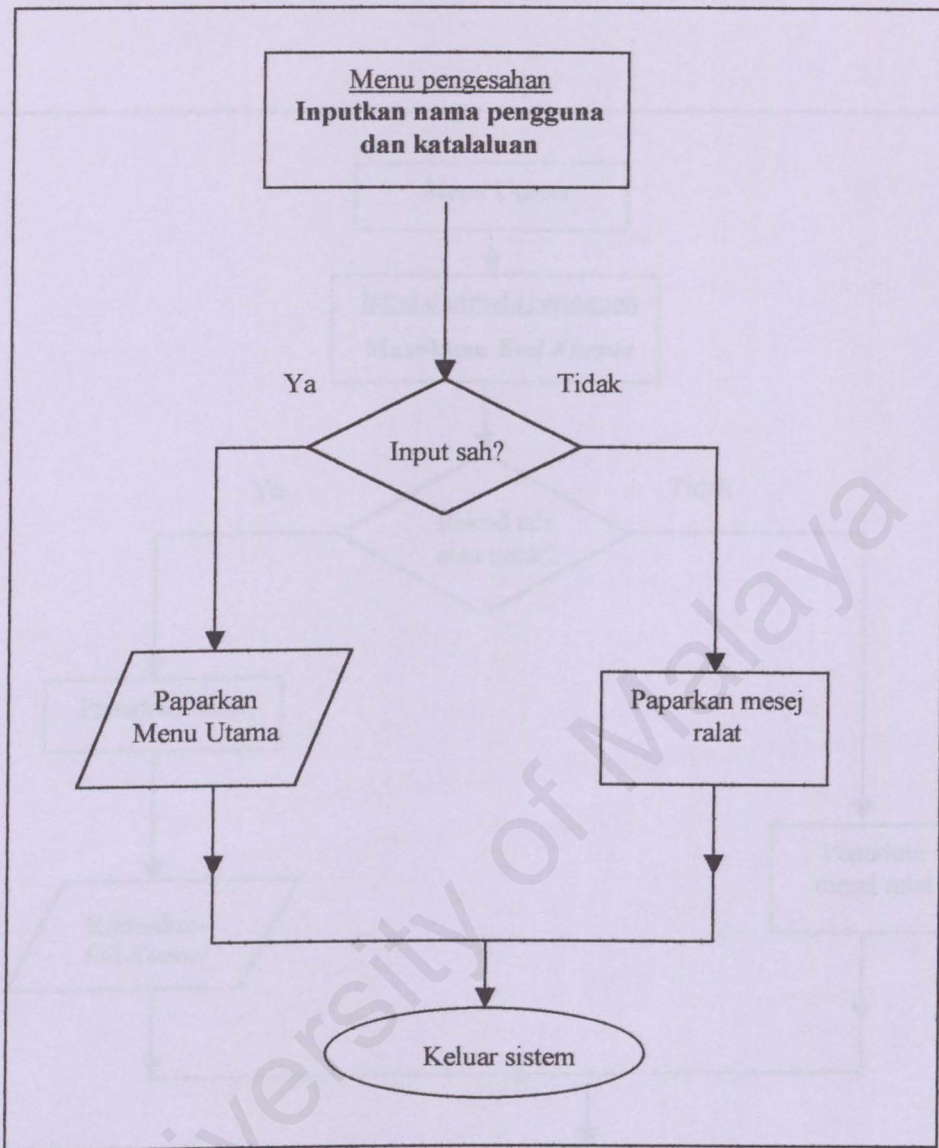
**Rajah 4.2 Pergerakan menu dalam SAP**

Berdasarkan rajah menu utama di atas, pengguna akan memasukkan pilihan kod kursus melalui *Menu Utama* sebagai pengesahan capaian ke atas rekod sistem. Seterusnya, pengguna akan memilih untuk melihat rekod pelajar atau melihat rekod keputusan ujian dan peperiksaan dengan membuat pilihan menerusi *Menu Perkhidmatan*.



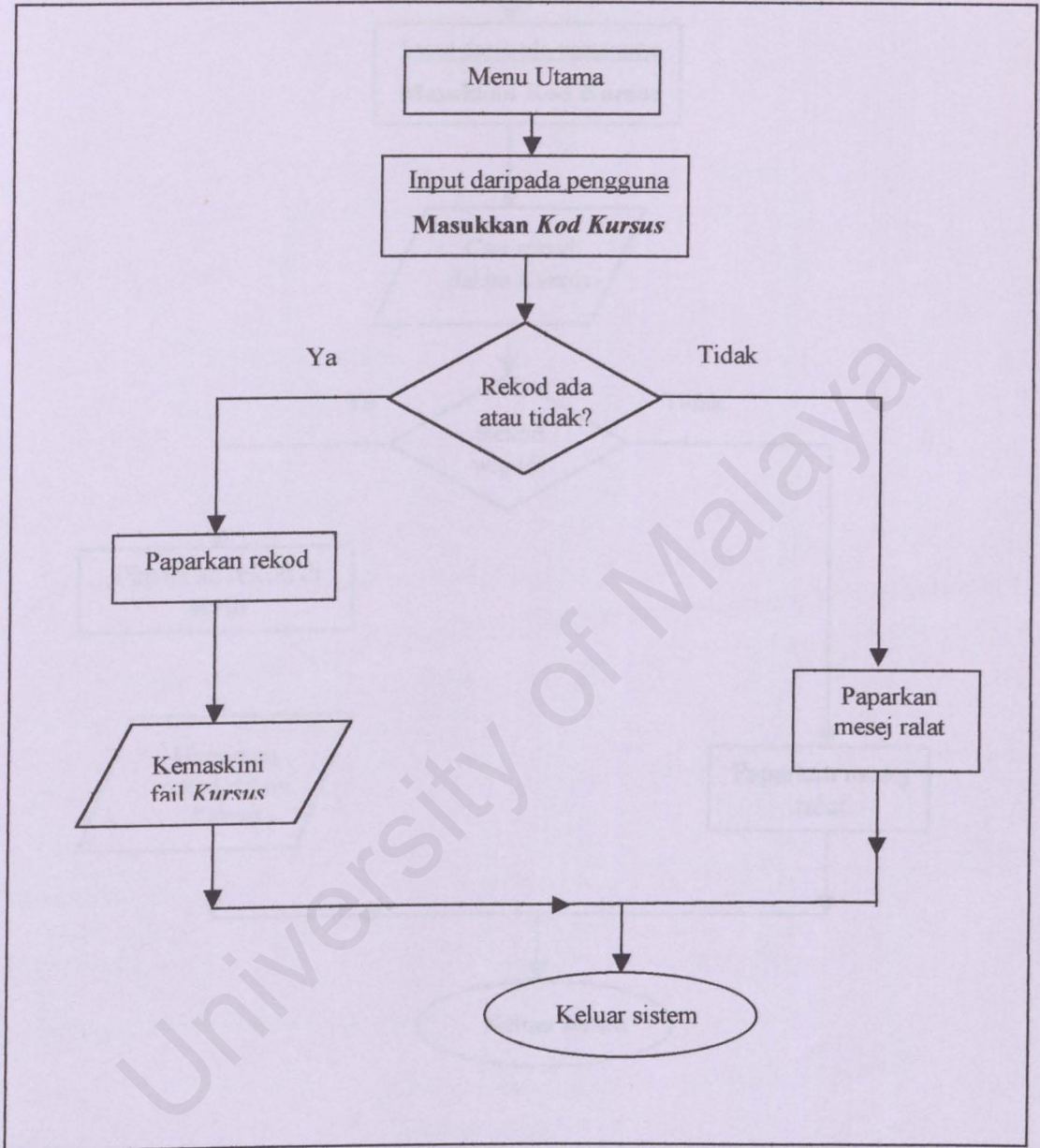
Rajah 4.3 : Pendaftaran masuk ke dalam SAP





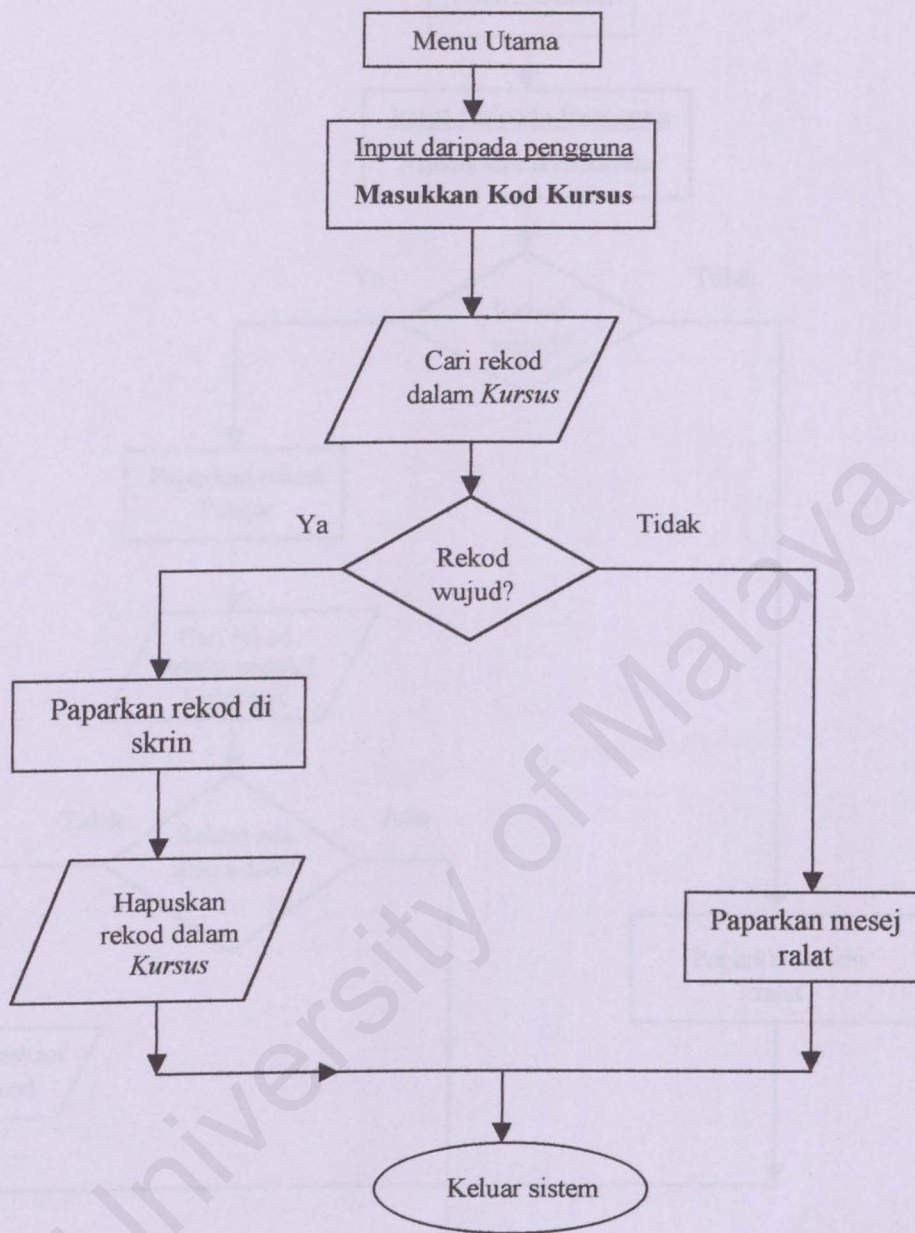
**Rajah 4.4 : Capaian ke Menu Utama selepas pengesahan nama pengguna dan katalaluan**

Carta-carta yang seterusnya menerangkan bagaimana proses-proses menambah, menghapus dan memilih format laporan dijalankan.

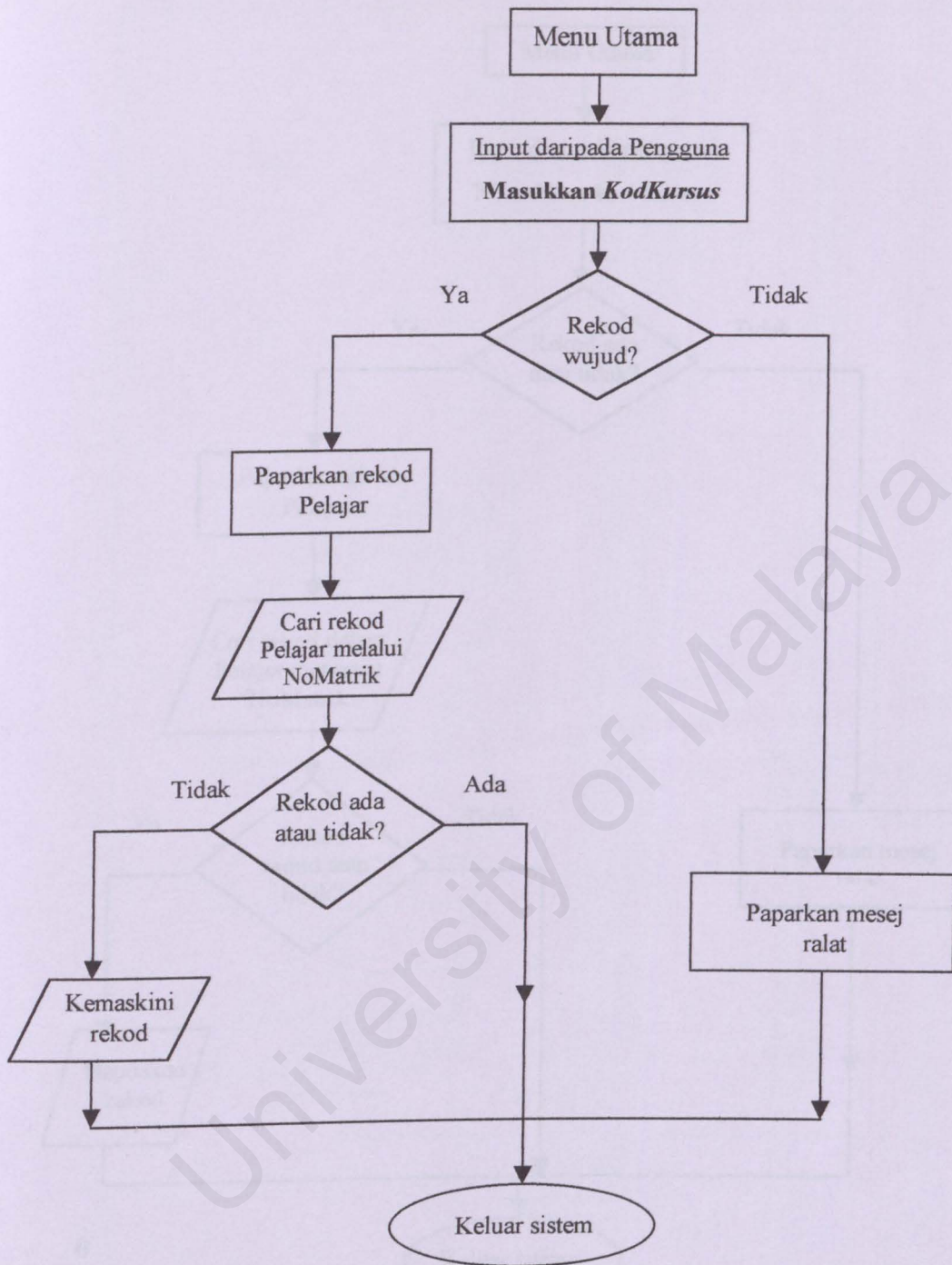


Rajah 4.5 : Operasi menambah rekod *Kursus* baru



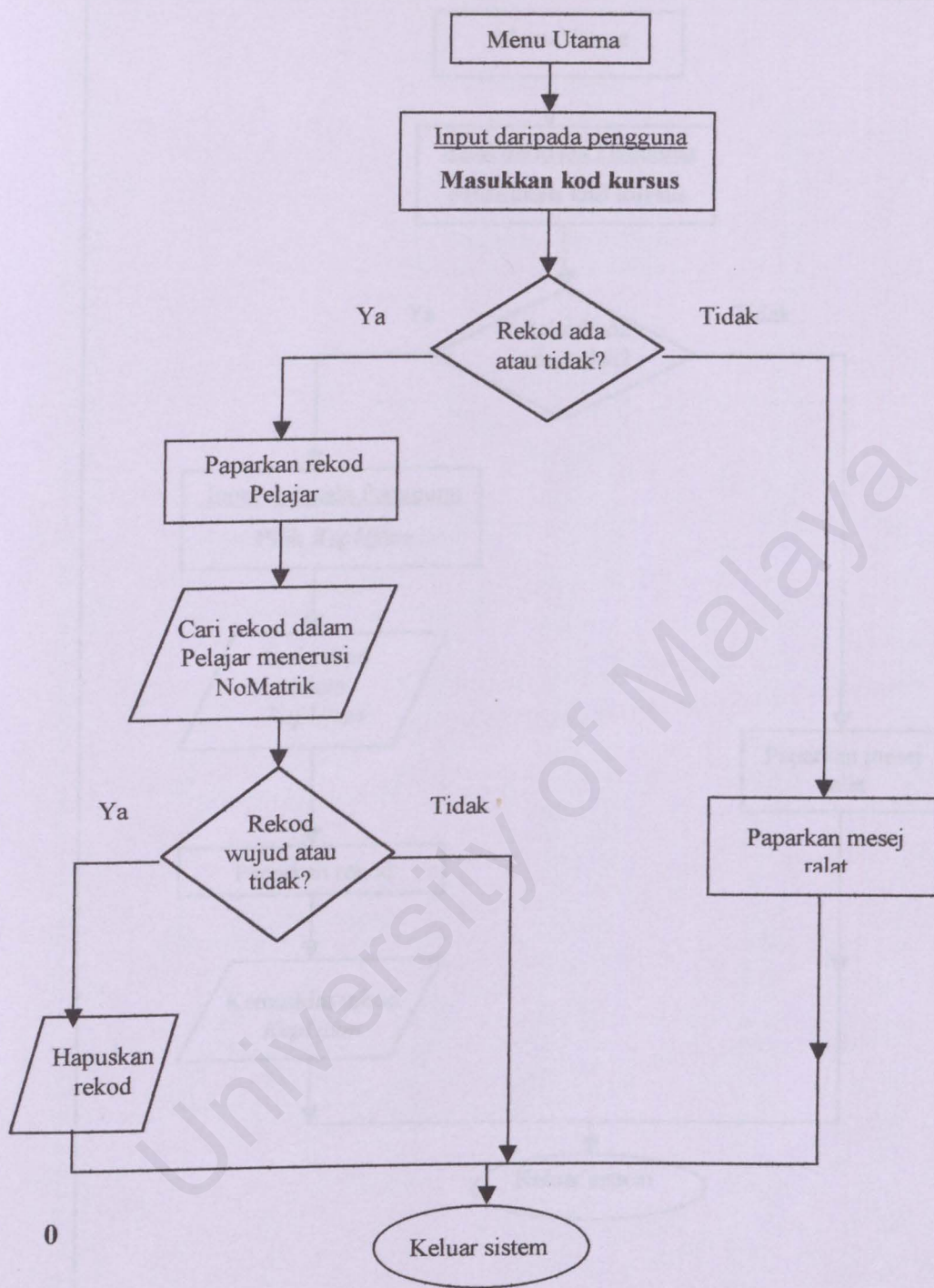


**Rajah 4.6 : Operasi menghapus rekod *Kursus***

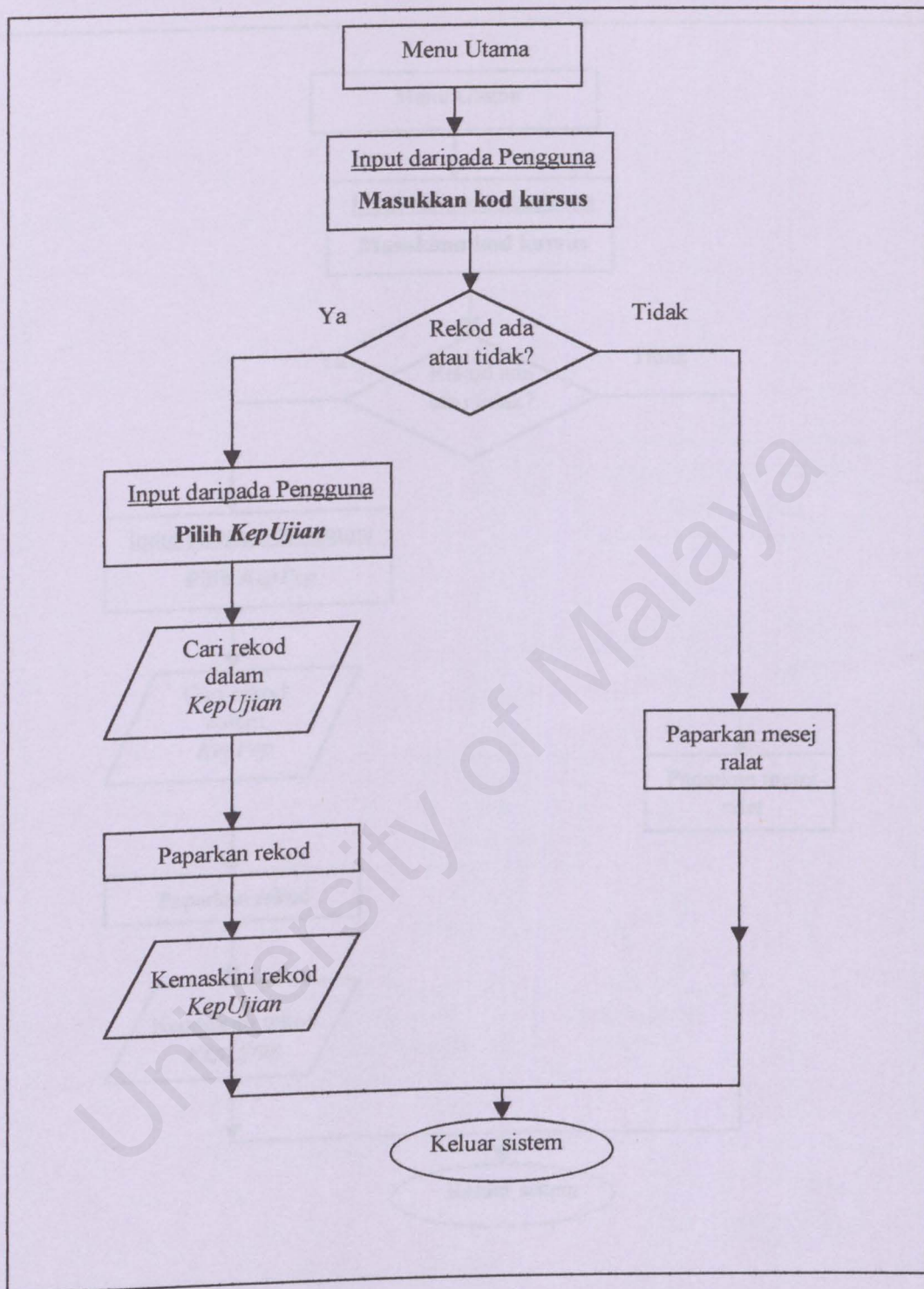


**Rajah 4.7 : Operasi menambah rekod *Pelajar***



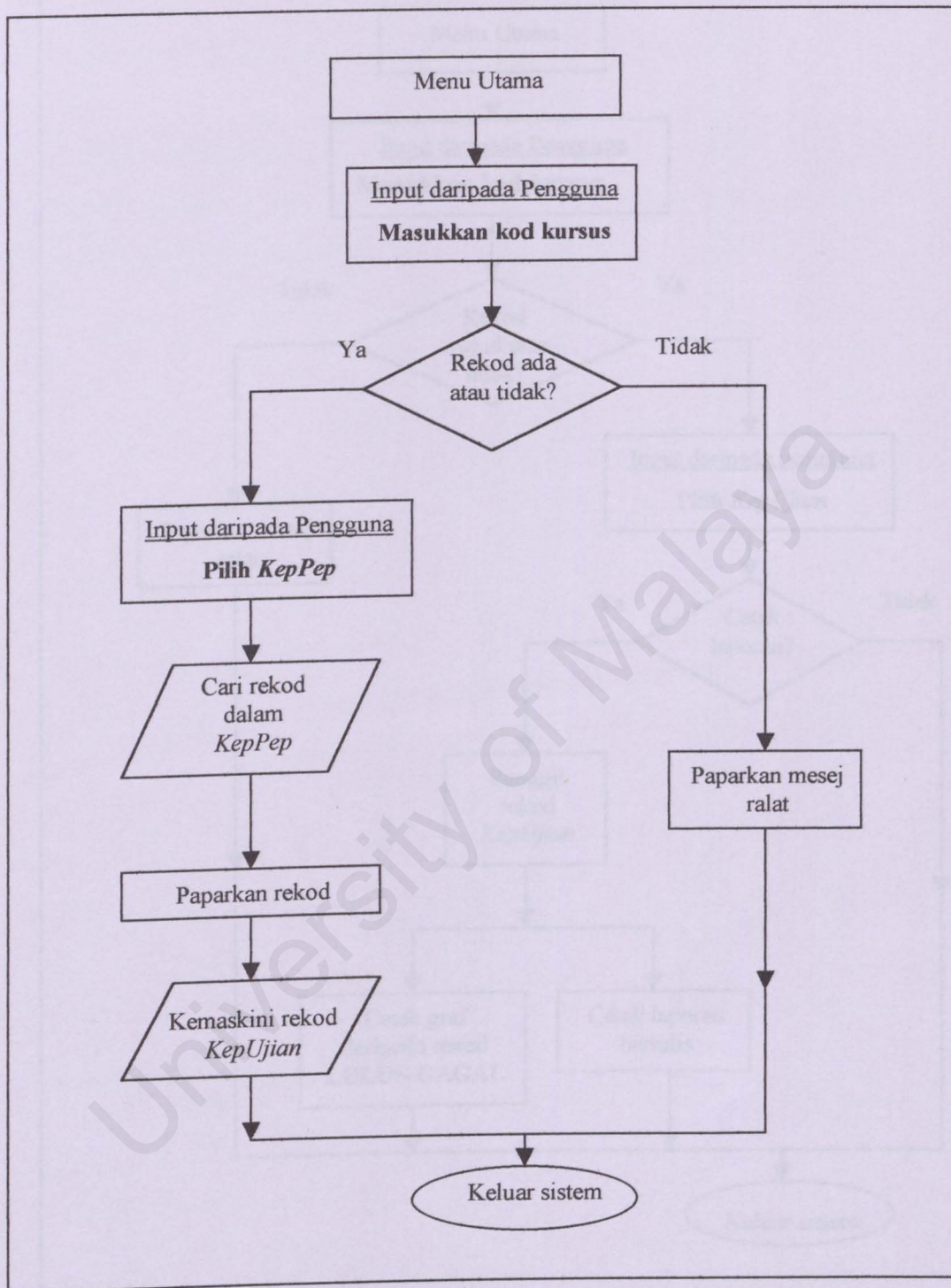


**Rajah 4.8 : Operasi menghapus rekod dalam *Pelajar***

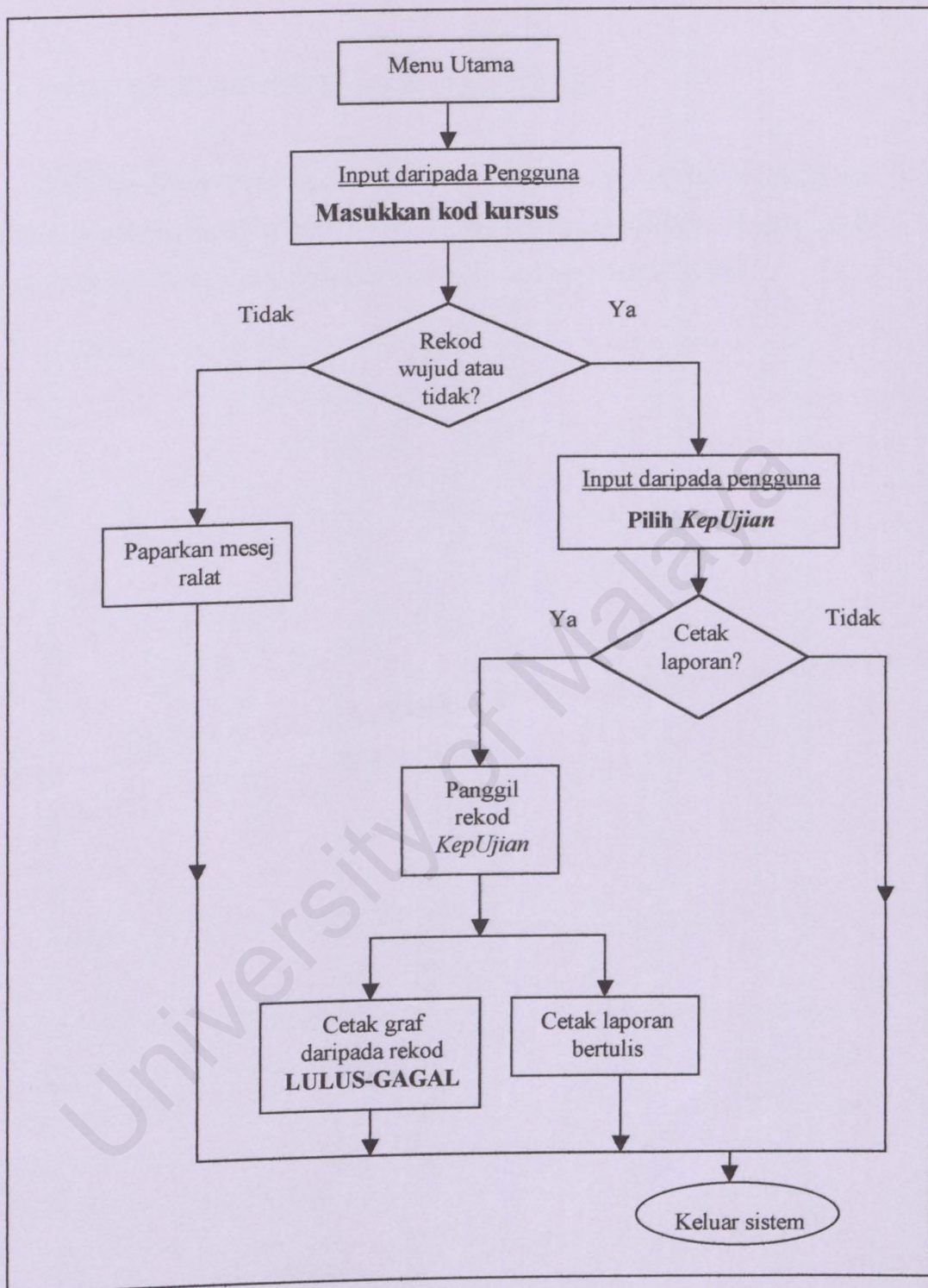


**Rajah 4.9 : Operasi menambah dan menghapus dalam *KepUjian***



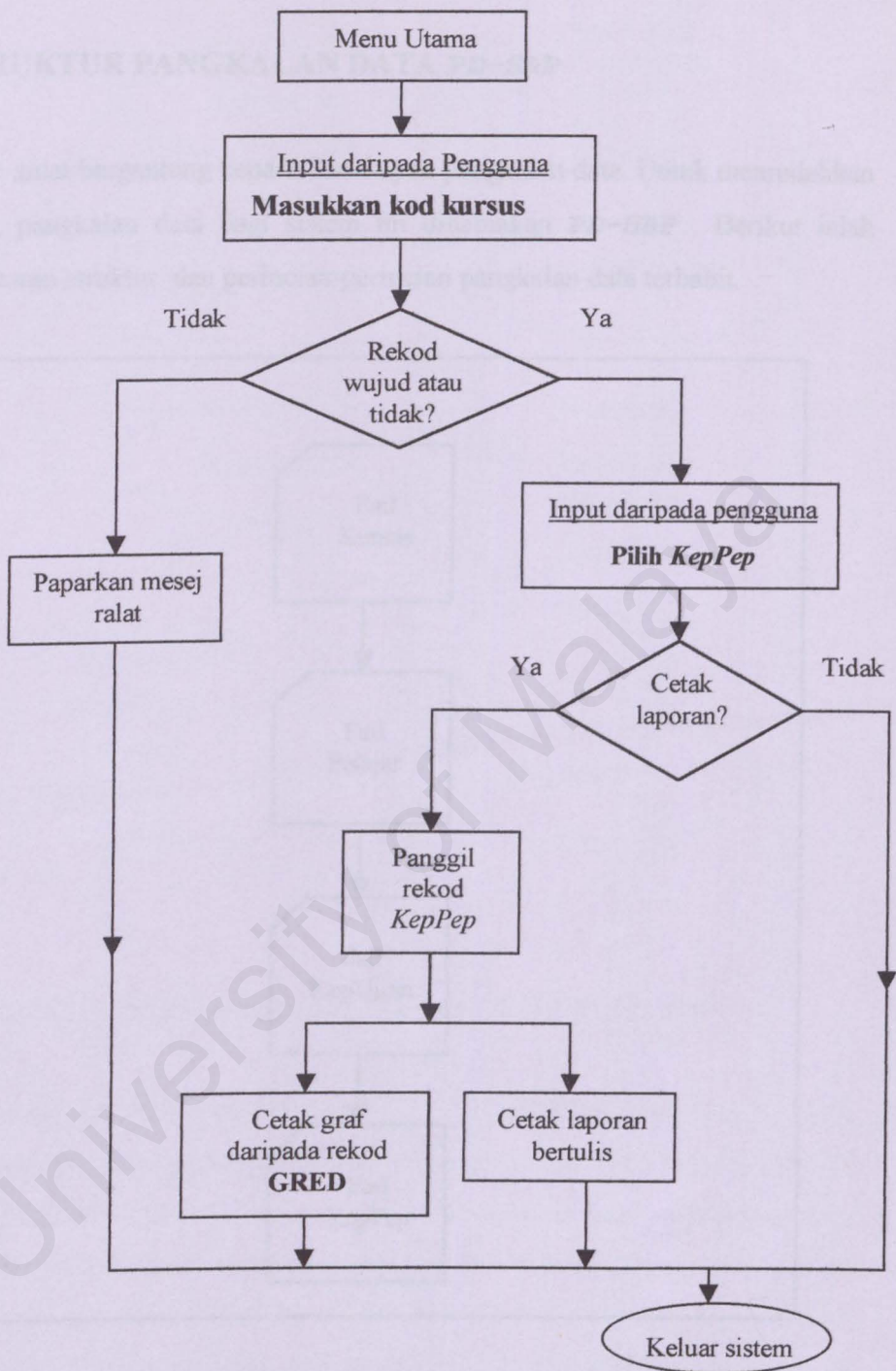


Rajah 4.10 : Operasi menambah dan menghapus dalam *KepPep*



Rajah 4.11 : Operasi mencetak output akhir dari *KepUjian*

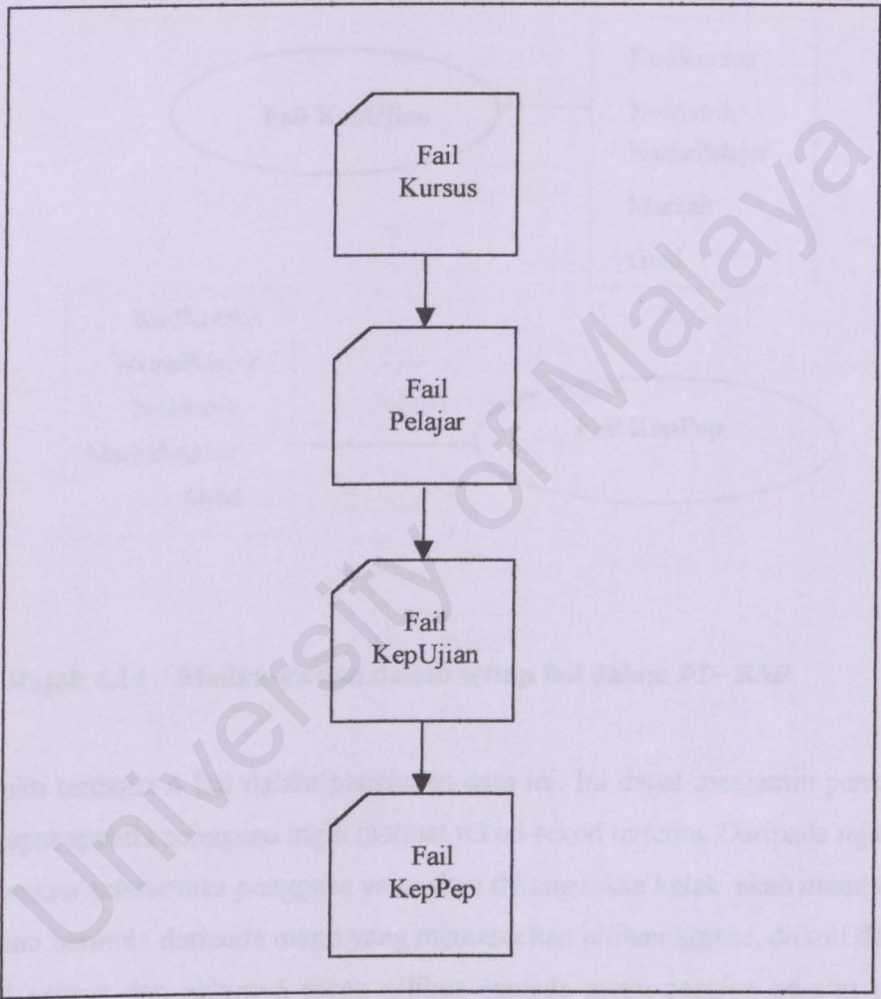




Rajah 4.12 : Operasi mencetak output akhir dari rekod *KepPep*

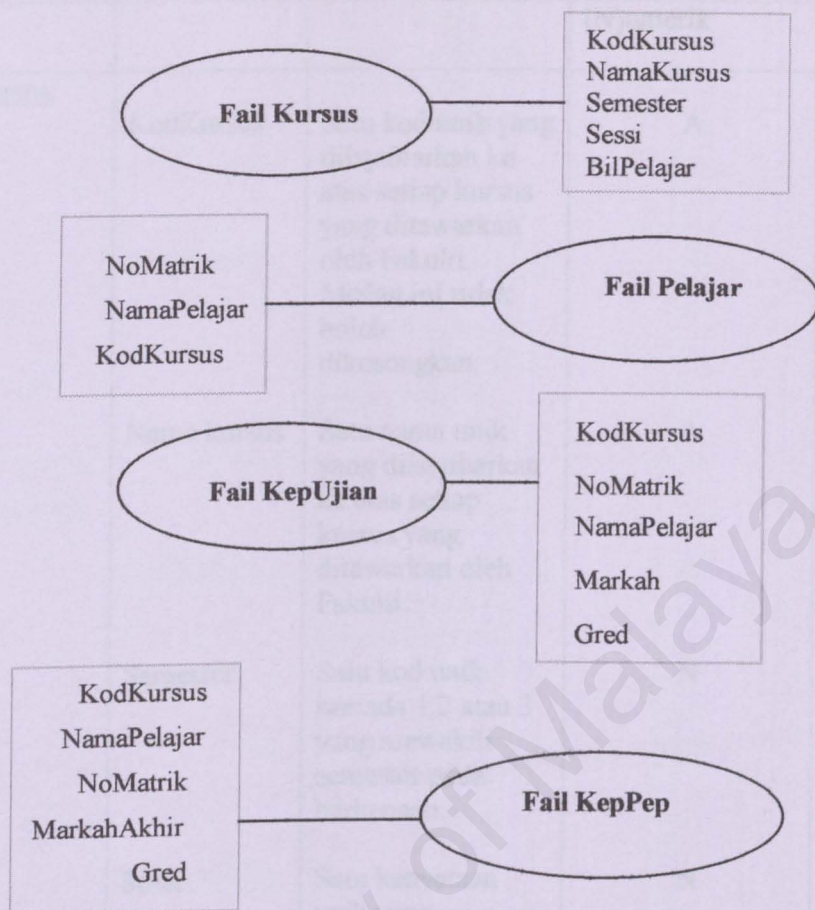
4.2 STRUKTUR PANGKALAN DATA *PD-SAP*

*SAP* amat bergantung kepada kecekapan pangkalan data. Untuk memudahkan penerangan, pangkalan data bagi sistem ini dinamakan *PD-SAP* . Berikut ialah adalah gambaran struktur dan perincian-perincian pangkalan data terbabit.



Rajah 4.13 Struktur fail *PD-SAP*





**Rajah 4.14 Medan-medan dalam setiap fail dalam PD-SAP**

Didapati terdapat 4 fail dalam pangkalan data ini. Ini dapat menjamin pencarian rekod yang cepat apabila pengguna ingin melihat rekod-rekod tertentu. Daripada rajah ini, dapat dibayangkan antaramuka pengguna yang akan dibangunkan kelak akan mempunyai 4 tahap capaian bermula daripada menu yang memaparkan pilihan kursus, diikuti dengan menu rekod pelajar dan akhirnya menu pilihan samada menu capaian ke atas rekod keputusan ujian atau rekod keputusan peperiksaan. Berikut adalah perincian setiap fail yang berkenaan:

**Jadual 4.2 Kamus data bagi PD-SAP**

No.	Nama fail	Nama medan	Keterangan	Jenis Medan [A]lfanumerik/	Panjang medan
-----	-----------	------------	------------	-------------------------------	---------------

				[N]umerik	
1.	<u>Fail Kursus:</u>	KodKursus	Satu kod unik yang diisytiharkan ke atas setiap kursus yang ditawarkan oleh Fakulti. Medan ini tidak boleh dikosongkan.	A	8
2.		Nama kursus	Satu nama unik yang diisytiharkan ke atas setiap kursus yang ditawarkan oleh Fakulti.	A	50
3.		Semester	Satu kod unik samada 1,2 atau 3 yang mewakili semester pada berkenaan.	N	1
4.		Sessi	Satu kenyataan unik yang mewakili sessi pengajian pada masa berkenaan.	N	9
5.		BilPelajar	Bilangan pelajar yang mendaftar bagi satu-satu kursus tersebut.	N	4
1.	<u>Fail Pelajar:</u>	NamaPelajar	Satu nama unik yang dimiliki oleh setiap Pelajar yang mendaftar mana-mana kursus di Fakulti.	A	100
2.		NoMatrik	Satu kod unik yang dimiliki oleh setiap	A	10



3.		KodKursus	<p>Pelajar yang mendaftar mana-mana kursus di Fakulti. Medan ini tidak boleh dikosongkan.</p> <p>Satu kod unik yang diisytiharkan ke atas setiap kursus yang diambil oleh setiap Pelajar.</p>	A	8
1.	<u>Fail KepUjian:</u>	KodKursus	Satu kod unik yang diisytiharkan ke atas setiap kursus yang diambil oleh setiap Pelajar berkenaan. Medan ini tidak boleh dikosongkan.	A	8
2.		NamaPelajar	Satu nama unik yang dimiliki oleh setiap Pelajar yang mendaftar mana-mana kursus di Fakulti.	A	100
3.		NoMatrik	Satu kod unik yang dimiliki oleh setiap Pelajar yang mendaftar Kursus tersebut. Medan ini tidak boleh dikosongkan.	A	10
4.		Markah	Jumlah markah yang diperolehi oleh setiap Pelajar berkenaan dalam Ujian.	N	2
5.		Gred	Penyataan samada	A	1

			Pelajar [L]ulus atau [G]agal bagi Ujian bagi Kursus berkenaan.		
1.	<u>Fail KepPep:</u>	KodKursus	Satu kod unik yang diisytiharkan ke atas setiap kursus yang diambil oleh setiap Pelajar berkenaan. Tidak boleh dikosongkan.	A	8
2.		NamaPelajar	Satu nama unik yang dimiliki oleh setiap Pelajar yang mendaftar mana-mana kursus di Fakulti.	A	100
3.		NoMatrik	Satu kod unik yang dimiliki oleh setiap Pelajar yang mendaftar Kursus tersebut. Medan ini tidak boleh dikosongkan.	A	10
4.		MarkahAkhir	Jumlah markah akhir yang diperolehi oleh Pelajar bagi kertas Peperiksaannya bagi Kursus tersebut.	N	2
5.		Gred	Satu kod unik yang diberikan kepada	A	1



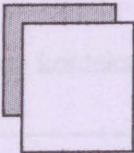

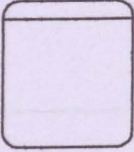
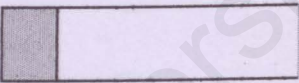
			setiap julat Markah tertentu. Dijanakan secara automatik oleh <b>SAP</b> dan disimpan.		
--	--	--	--	--	--



4.3 GAMBARAJAH ALIRAN DATA (DFD)

4.3 GAMBARAJAH ALIRAN DATA (DFD)

Gambarajah aliran data (*Data Flow Diagram-DFD*) ialah persembahan secara grafik penyimpanan aliran data di dalam sistem. DFD adalah struktur analisis dan alatan rekabentuk yang membenarkan penganalisa untuk memahami sistem dan subsistem sebagai satu set aliran data yang saling berhubungan. DFD menunjukkan input, proses serta output yang berlaku di dalam sistem. DFD yang akan digunakan dalam penerangan seterusnya adalah menggunakan pendekatan Gane dan Sarson.

Simbol	Penerangan
	Entiti
	Aliran data
	Proses
	Penyimpanan data

Jadual 4.3 Simbol dalam gambarajah aliran data

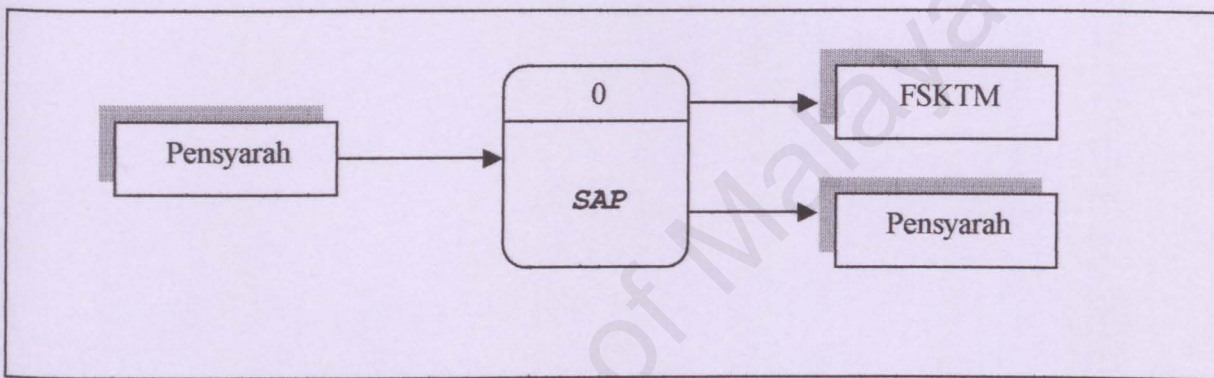
#### 4.3.1 GAMBARAJAH ALIRAN DATA BAGI SAP

DFD akan dibahagikan kepada 3 bahagian iaitu diagram konteks, diagram '0' dan diagram anak.



- ❑ Diagram konteks menerangkan tahap tertinggi aliran data yang menunjukkan hanya satu proses yang mewakili sistem secara keseluruhannya. Hanya melibatkan entiti dan proses.
- ❑ Diagram '0' merupakan capahan daripada diagram konteks yang memecahkan satu 'proses' tadi kepada proses-proses yang lebih spesifik ( sehingga 9 proses). Ia turut memasukkan entiti, storan data dan proses secara spesifik.
- ❑ Diagram anak merupakan gambaran aliran data yang paling spesifik mewakili setiap proses yang ada.

Berikut adalah diagram konteks bagi **SAP**:



**Rajah 4.15 Diagram konteks SAP**

Seterusnya ialah diagram '0':

Diagram anak yang merupakan kembangan bagi setiap proses dalam Diagram '0' dapat dilihat di Lampiran A.



#### 4.4 GAMBARAN ANTARAMUKA PENGGUNA *SAP*

Bagi kebanyakan pengguna; terutama bagi individu bukan pengaturcara mahupun pembangun sistem, antaramuka pengguna adalah 'sistem'. Antaramuka membantu

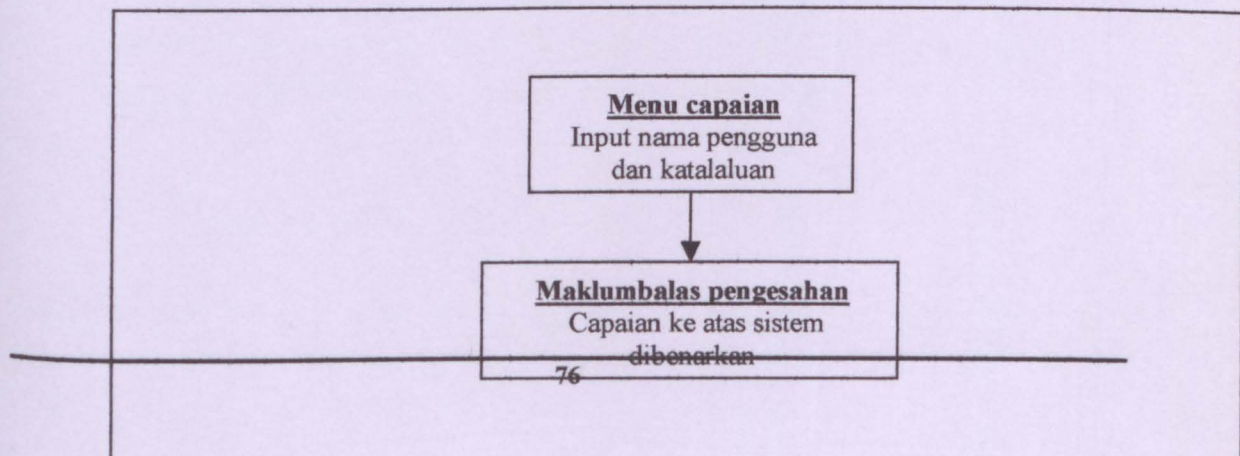
interaksi di antara pengguna dengan sistem. Antaramuka pengguna terbahagi kepada dua komponen iaitu bahasa persembahan (daripada sistem-kepada-pengguna) dan bahasa tindakan (daripada pengguna-kepada-sistem).

Berikut adalah perkara-perkara yang perlu dipertimbangkan apabila merekabentuk antaramuka pengguna:

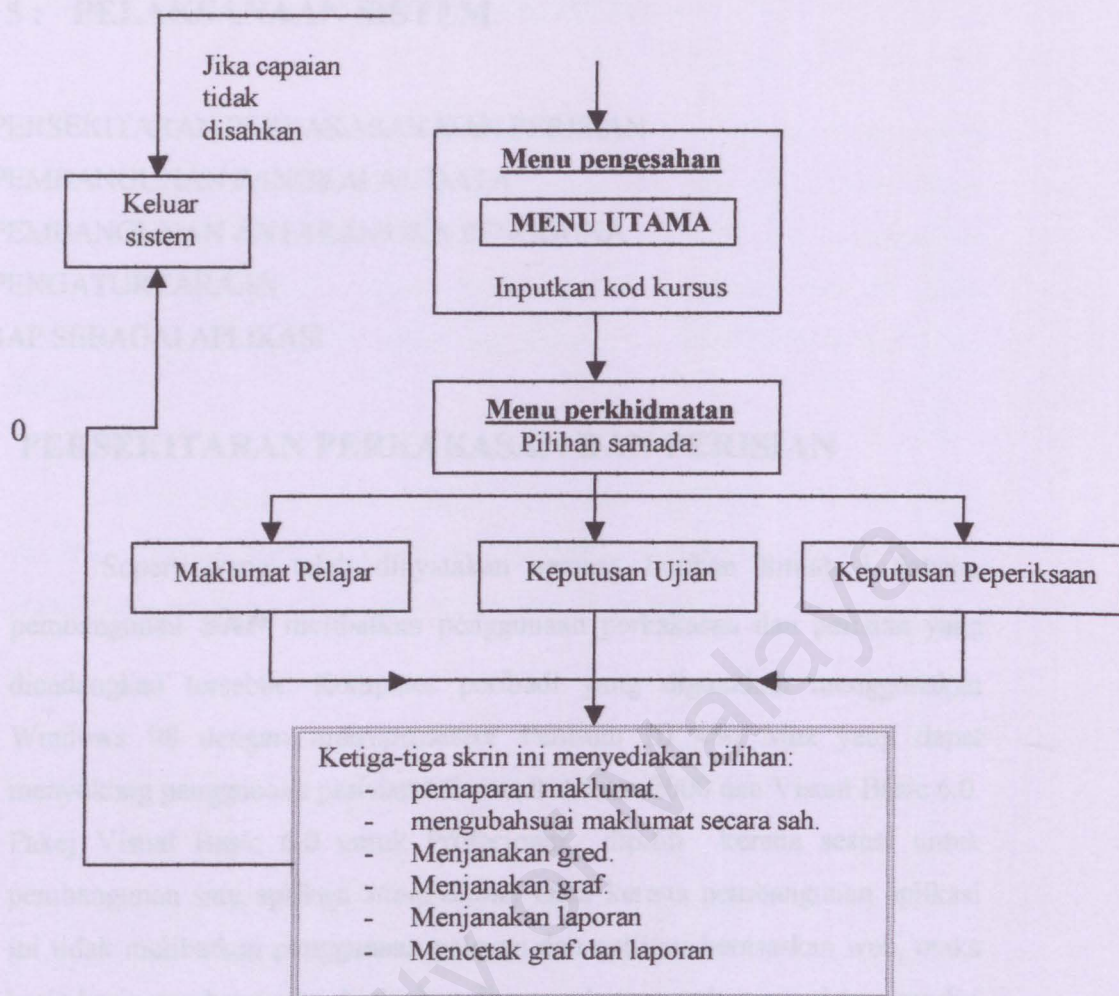
- 1) Keberkesanan rekabentuk antaramuka dalam membolehkan pengguna mencapai dan menggunakan sistem untuk mencapai matlamatnya.
- 2) Keberkesanan rekabentuk antaramuka dalam meningkatkan kelajuan masukan data dan mengurangkan ralat.
- 3) Merekabentuk antaramuka yang mampu memberi maklum balas yang tepat dan diperlukan kepada pengguna.
- 4) Samada antaramuka yang direkabentuk dapat meningkatkan produktiviti pengguna dan memperbaiki kondisi pekerjaannya.

Oleh itu, antaramuka yang baik harus mudah difahami melalui penggunaan bahasa umum yang diterima oleh majoriti pengguna. Terdapat beberapa jenis antaramuka pengguna; contohnya antaramuka pengguna bergrafik (*GUI-Graphical User Interfaces*), menu, antaramuka berasaskan borang, antaramuka bahasa tabii dan antaramuka berasaskan Web.

Antaramuka bagi **SAP** akan menggunakan pendekatan bergrafik (GUI) berasaskan Windows yang akan dibangunkan menggunakan Visual Basic 6.0. Rajah berikut menunjukkan dokumentasi rekabentuk antaramuka bagi **SAP**.







**Rajah 4.17** Rekabentuk antaramuka pengguna **SAP**

## BAB 5 : PELAKSANAAN SISTEM

### 5.1 PERSEKITARAN PERKAKASAN DAN PERISIAN

### 5.2 PEMBANGUNAN PANGKALAN DATA

### 5.3 PEMBANGUNAN ANTARAMUKA PENGGUNA

### 5.4 PENGATURCARAAN

### 5.5 SAP SEBAGAI APLIKASI

## 5.1 PERSEKITARAN PERKAKASAN DAN PERISIAN

Seperti yang telah dinyatakan semasa Latihan Ilmiah I dahulu, pembangunan **SAP** melibatkan penggunaan perkakasan dan perisian yang dicadangkan tersebut. Komputer peribadi yang digunakan menggunakan Windows 98 dengan mikroprosesor Pentium III 450 Mhz yang dapat menyokong penggunaan perisian Microsoft Access 2000 dan Visual Basic 6.0. Pakej Visual Basic 6.0 untuk Profesional dipilih kerana sesuai untuk pembangunan satu aplikasi *stand-alone*. Oleh kerana pembangunan aplikasi ini tidak melibatkan penggunaan pelayan dan aplikasi berasaskan web, maka kerja-kerja pembangunan dijalankan di ruang kerja pembangun sistem sendiri (kolej kediaman).

Pembangunan **SAP** dimulakan pada penghujung semester pertama, iaitu sebaik sahaja kertas cadangan projek disahkan. Keseluruhan pembangunan telah mengambil 15 minggu termasuk pembangunan pangkalan data, antaramuka pengguna, implementasi dan penyelenggaraan dan penyediaan laporan akhir projek ini.



## 5.2 PEMBANGUNAN PANGKALAN DATA

Pangkalan data ialah komponen pertama yang diwujudkan kerana sedikit sebanyak ia membantu kita lebih memahami skop dan domain sistem. Pangkalan data akhirnya dikenali sebagai **PD-SAP** ini dibangunkan menggunakan Microsoft Access 2000. Struktur pangkalan data ini dapat diperolehi dari Bahagian 4.2 dalam bab keempat *STRUKTUR PANGKALAN DATA SAP*.

Walau bagaimanapun terdapat juga sedikit perubahan pada sesetengah strukturnya untuk disesuaikan dengan keperluan sistem kelak, terutama setelah antaramuka pengguna direka dan pengkodan dijalankan.

## 5.3 PEMBANGUNAN ANTARAMUKA PENGGUNA

Antaramuka pengguna dibangunkan menggunakan perisian Visual Basic 6.0 edisi Profesional yang menyediakan banyak ciri-ciri menarik dalam merekabentuk rupa borang (*form*), menentukan *event*, panduan mengkod dan komponen-komponen untuk memudahkan pembangunan satu-satu sistem kecil.

Perisian ini menyediakan beberapa pilihan untuk memulakan pembangunan sistem; samada bermula dari *wizard* seperti VB Application Wizard atau terus memulakan projek tanpa bantuan *wizard*. Selain itu, beberapa format projek boleh dipilih. *Standard Exe* memberikan persekitaran untuk pembangunan aplikasi berasaskan Windows yang biasa, maka ia telah dipilih untuk pembangunan SAP. Rupa persekitaran perisian ini dikenali sebagai VB IDE atau *Integrated Development Environment* yang

mewujudkan satu persekitaran pengaturcaraan visual yang menarik dan berstruktur. VB memudahkan rekabentuk antaramuka sekaligus menghidupkan skrin tersebut melalui pengkodan.

Komponen *Data Environment Designer* telah digunakan untuk menghubungkan pangkalan data ke skrin, set objek dan perhubungan antara objek-objek dan menjana laporan. Senarai maklumat pelajar, laporan keputusan Ujian dan Peperiksaan dihasilkan dengan mudah menerusi penggunaan komponen ini. Sementara itu, skrin yang memaparkan maklumat kursus, pelajar, keputusan Ujian dan Peperiksaan direka menggunakan kombinasi *Data control* dan *Data Aware Control*.

## 5.4 PENGATURCARAAN

Meskipun VB memberikan satu rekabentuk antaramuka yang menarik, skrin itu tidak akan menjadi aktif tanpa pengaturcaraan. Matlamat pengaturcaraan menggunakan VB ialah untuk menyediakan satu set kod bebas yang akan diaktifkan apabila sesuatu event berlaku (disesuaikan dengan sifat kod yang setiap objek miliki). Bagi setiap *control* (seperti butang, kotak teks atau imej) yang digunakan di atas setiap skrin VB, sedikit kod tetap diperlukan untuk mengaktifkan objek.

Contohnya, dalam **SAP** sendiri, bagi satu butang yang dinamakan `cmdMenuUtama` serta membawa *caption* bertulis "&Menu Utama" dan digunakan untuk mengklik untuk ke menu utama, kod aturcaranya adalah:

```
Private Sub cmdMenuUtama_Click()
```



```
frmExit.Show
```

```
Me.Hide
```

```
End Sub
```

di mana ia menunjukkan jika pengguna mengklik sekali pada butang "Menu Utama", skrin menu utama akan dipaparkan dan skrin yang berkenaan akan *deactivate* dan disembunyikan daripada skrin utama.

Kod dibawah adalah kod yang telah diletakkan pada skrin tamat (exit) bagi **SAP**:

```
Private Sub Form_Load()
```

```
    frmExit.Show
```

```
    DrawWidth = 4
```

```
    For i = 1 To 20000
```

```
        Down = Down + 1
```

```
        Across = Across + 1
```

```
        PSet (Rnd * Across, Rnd * Down), QBColor(Rnd * 15)
```

```
    Next i
```

```
    JamRandik.Enabled = True
```

```
    JamRandik.Interval = 1500
```

```
End
```

```
End Sub
```

Kod di atas menggambarkan apa yang akan berlaku apabila skrin tamat diaktifkan ( *Form\_Load*). Dalam sela masa *interval* 1500, skrin tamat ini akan dipaparkan serentak dengan *random dots* dalam pelbagai warna bertaburan dari atas ke bawah serta merentangi skrin. Kemudian, baru skrin ditutup dan pengguna dibawa keluar dari sistem.

Dalam sistem ini, pengaturcaraan VB digunakan sepenuhnya dalam menghubungkan skrin-ke-skrin, pangkalan data kepada skrin, penggredan dan penjanaan graf yang merupakan matlamat utama **SAP**.

## 5.5 SAP SEBAGAI APLIKASI

Pembangunan **SAP** dimulaakan dengan pangkalan data menggunakan Access 2000, diikuti rekabentuk skrin-skrin menggunakan VB 6.0. Untuk menyempurnakan **SAP** sebagai satu aplikasi, pengkodan diperlukan untuk mengintegrasikan kedua-kedua komponen ini. Proses ini memakan masa hampir dua bulan.

Fasa penggredan dan penjanaan graf merupakan fasa paling mencabar dan memerlukan teknik pengaturcaraan yang rumit. Pembangun sistem harus memastikan penggredan tepat samada untuk ujian atau peperiksaan dn graf dan laporan serta graf yang dijanakan sentiasa mengikut perubahan yang berlaku dalam pangkalan data.



## BAB 6 : PENGUJIAN SISTEM

### 6.1 PENGUJIAN TAHAP DEMI TAHAP

### 6.2 PENYELENGARAAN SISTEM

### 6.1 PENGUJIAN TAHAP DEMI TAHAP

Pengujian merupakan fasa yang paling penting setelah sesuatu sistem berjaya diwujudkan. Bagi **SAP**, pengujian dilakukan tahap demi tahap untuk memastikan setiap unit dalam sistem berfungsi sewajarnya.

#### 6.1.1 PENGUJIAN KATA LALUAN

Pengujian kata laluan sangat penting kerana ia menentukan tahap kebolehpercayaan sistem. Pengguna yang baru pertama kali menggunakan sistem diminta mendaftarkan nama pengguna dan katalaluan yang unik untuk membolehkannya memasuki sistem (walaupun sistem bersifat *stand-alone*).

Pada awalnya, pembangun sistem berhasrat mengkod bahagian katalaluan ini supaya sistem ini menggunakan nama pengguna dan katalaluan Windows (dengan mengandaikan majoriti pengguna komputer mempunyai katalaluan Windows sendiri). Walau bagaimanapun, kaedah ini tidak sesuai.

Sessi pengujian melibatkan beberapa kali capaian ke atas sistem untuk memastikan hanya namap pengguna dan katalaluan yang diterima sistem sahaja dianggap sah untuk melihat rekod yang disimpan dalam sistem.

#### 6.1.2 PENGUJIAN KOD KURSUS

Pengujian ini memerlukan pengguna memilih kod kursus yang ada dalam simpanan pangkalan data untuk membolehkannya melihat rekod pelajar dan keputusan akademik bagi kursus itu sahaja.

### **6.1.3 PENGUJIAN PANGKALAN DATA**

Pengujian ini memerlukan pangkalan data sentiasa berada dalam keadaan terkini dengan memastikan hanya maklumat yang berkenaan sahaja dipaparkan kepada pengguna. Pengkodan digunakan untuk memastikan pangkalan data berfungsi dengan tepat setiap kali penambahan, penghapusan, kemaskini dan pewujudan rekod baru berlaku.

### **6.1.4 PENGUJIAN PENJANAAN GRED**

Pengujian perlu memastikan apabila butang 'Jana Gred' diklik oleh pengguna samada pada skrin rekod Keputusan Ujian atau Peperiksaan, gred yang berpadanan dengan markah yang dipaparkan bagi setiap pelajar akan dijanakan secara langsung ke dalam pangkalan data.

### **6.1.5 PENGUJIAN PENJANAAN GRAF**

Pengujian perlu memastikan apabila butang 'Jana Graf' diklik oleh pengguna samada pada bagi rekod Keputusan Ujian atau Peperiksaan, graf yang dijanakan mesti berpadanan dengan rekod terkini dari pangkalan data.

### **6.1.6 PENGUJIAN LAPORAN**



Pengujian ini penting untuk memastikan laporan yang boleh dicetak (samaada laporan maklumat pelajar, keputusan ujian, keputusan peperiksaan mahupun graf pencapaian) dijanakan berpadanan dengan rekod dari pangkalan data yang terkini.

## 6.2 PENYELENGGARAAN SISTEM

Output yang diperolehi daripada bahagian pengujian membolehkan pembangun sistem memperbaiki sistemnya supaya matlamat pembangunannya tercapai. Setelah menjalankan beberapa kali pengujian pada setiap tahap yang telah dihuraikan tadi, banyak ralat dan ketidakcekapan dikenalpasti.

Oleh itu, beberapa pengubahsuaian dan pembaikan dilakukan untuk memulihkan sistem, terutama dalam bahagian-bahagian yang melibatkan pengaturcaraan.

## **BAB 7: PENILAIAN SISTEM**

- 7.1 PENILAIAN KESELURUHAN SISTEM
- 7.2 MASALAH-MASALAH YANG DIHADAPI
- 7.3 KELEBIHAN DAN KELEMAHAN SISTEM
- 7.4 KESIMPULAN

### **7.1 PENILAIAN KESELURUHAN SISTEM**

Setelah menjalankan beberapa kali sesi pengujian (termasuk ketika VIVA), didapati sistem yang dibangunkan masih menunjukkan beberapa kekurangan. Kekurangan paling ketara adalah pada penjanaan graf, disebabkan ketidakcekapan pengaturcaraan. Selain itu, terdapat juga beberapa dalam penggredan markah peperiksaan di mana terdapat beberapa sampel markah diberikan gred yang tidak munasabah. Contohnya, terdapat markah peperiksaan bernilai 80 diberikan gred D (gred yang sepatutnya adalah 'A').

Oleh itu, sistem ini memerlukan pemerhatian semula terutama pada bahagian pengaturcaraan supaya ralat-ralat diatas dapat dielakkan. Ini penting untuk memastikan sistem dapat mencapai matlamat utamanya iaitu penjanaan gred dan graf yang tepat.

### **7.2 MASALAH-MASALAH YANG DIHADAPI**



### 7.2.1 MASALAH PENGATURCARAAN

Kemahiran pengaturcaraan menggunakan Visual Basic 6.0 amat penting, terutama dalam penjana gred dan graf. Oleh kerana pembangun sistem tidak berpengalaman menggunakan pengaturcaraan VB secara intensif, maka banyak rujukan di Internet dan contoh-contoh tesis terdahulu telah dilakukan untuk berusaha memahami teknik pengaturcaraan.

Walaupun VB 6.0 menyediakan beberapa komponen yang memudahkan kerja-kerja diatas (seperti Microsoft Chart Control 6.0 untuk menjana graf), kod yang tepat perlu diketahui supaya setiap *control* dan komponen VB yang digunakan itu berfungsi. Penggredan yang tidak tepat, seperti yang telah disebut sebelum ini, disebabkan oleh penggunaan gelung (loop) dalam pengaturcaraan yang tidak cekap.

### 7.2.2 PERISIAN -PERISIAN TIDAK SERASI

Masalah juga dihadapi ketika mula-mula menghubungkan skrin kepada pangkalan data kerana kedua-dua perisian tidak serasi. Perisian VB 6.0 yang digunakan tidak dapat menyokong versi Access 2000. Pangkalan data yang telah dibangunkan itu akhirnya dibaiki dan diubahsuai (*repair and compact*) supaya dapat dihubungkan kepada skrin VB. Oleh itu, sebarang pengubahsuaian yang perlu dilakukan terhadap pangkalan data hanya boleh dilakukan melalui skrin VB kerana pangkalan data hanya boleh dilihat semula di Microsoft Access 2000 sebagai fail *read-only*.

## 7.3 KELEBIHAN DAN KELEMAHAN SISTEM

### 7.3.1 KELEBIHAN SISTEM

Sekiranya **SAP** berjaya disempurnakan tanpa sebarang ralat, ia akan menjadi satu sistem sokongan yang berkesan untuk kegunaan para pensyarah. Ia akan menjadi satu 'pusat' menyimpan rekod maklumat pelajar, rekod keputusan ujian dan peperiksaan dan memudahkan penggredan dan menjana graf pencapaian untuk tujuan menganalisis pencapaian kursus dan keberkesanan kursus.

Ia juga mudah digunakan kerana hanya melibatkan aktiviti mengklik butang-butang arahan (*keyboard shortcut* juga dibenarkan) pada skrin dan arahan-arahan pada skrin juga mudah difahami. Oleh kerana ia bersifat *stand-alone*, ia lebih bersifat persendirian untuk kegunaan peribadi pensyarah.

### 7.3.2 KELEMAHAN SISTEM

Kelemahan asas dilihat pada bahagian rekod Ujian. Bahagian ini hanya menyediakan masukan rekod untuk satu Ujian sahaja (dengan andaian hanya satu ujian *mid-term* sahaja dibuat bagi satu-satu kursus). Kadangkala, lebih dari satu Ujian dilakukan. Dalam kes ini, pengguna tiada pilihan untuk menambah rekod Ujian tambahan pada sistem.

Selain itu, keselamatan sistem ini bergantung sepenuhnya pada keselamatan komputer peribadi pengguna. Oleh itu, sewajarnya hanya seorang pengguna sahaja yang boleh menggunakan sistem ini. Maka, log-in pada awal sistem seharusnya Cuma boleh menerima satu pendaftaran sahaja; iaitu pada kali pertama pengguna hendak menggunakan sistem.



## 7.4 KESIMPULAN

Terdapat beberapa sebab mengapa **SAP** tidak dibangunkan secara atas talian mahupun secara rangkaian. Sebagai satu sistem kecil yang lebih bersifat dan individu, **SAP** lebih sesuai dijadikan sebagai sistem *stand-alone* di mana pengguna (pensyarah) akan menginstalasi **SAP** melalui perisian yang memuatkannya kepada komputer peribadi di pejabat masing-masing. Walaupun terdapat idea untuk meletakkan **SAP** dalam LAN FSKTM, didapati pihak Fakulti masih belum dapat menyediakan rangkaian yang dapat menghubungkan komputer peribadi sesama pensyarah sendiri.

Oleh itu, **SAP** hanya dibangunkan secara *stand-alone* untuk memudahkan kerja penggredan, menganalisis dan menyimpan maklumat keputusan ujian dan peperiksaan pelajar. Apa yang penting, sistem ini dapat melakukan penggredan dengan tepat dan menghasilkan graf untuk tujuan analisis bagi setiap kursus dalam kendalian satu-satu pensyarah. Kedua-dua ini adalah apa yang cuba hendak dicapai oleh **SAP** sebagai satu sistem sokongan kepada pensyarah.

## SENARAI RUJUKAN

- Kenneth E. Kendall and Julie E. Kendall. 1996. *System Analysis and Design*, 4<sup>th</sup> Edition, Prentice Hall Inc. New Jersey.
- Ian Sommerville. 1995. *Software Engineering*, 5<sup>th</sup> Edition, Addison Wesley Publishers Ltd. New York.
- H.M Deitel, P.J Deitel and T.R Deitel. 1999. *Visual Basic 6 : How to Program*. Prentice Hall Inc. New Jersey.
- D.M Kroenke. 1998. *Database Management: Fundamentals, Design and Implementation*, 6<sup>th</sup> Edition. Prentice Hall Inc. New Jersey.
- Tim Apps. 1996. *Practical Microsoft Access 97*. Prentice Hall Inc. Australia.
- System [online]. Available HTTP: URL [www.ed-soft.com](http://www.ed-soft.com).  
tarikh akses 2/7/2001.
- System [online]. Available HTTP: URL [www.cstp.umkc.edu/~cblade](http://www.cstp.umkc.edu/~cblade).  
tarikh akses 7/7/2001.
- System [online]. Available HTTP: URL [www.ndsu.edu/](http://www.ndsu.edu/) . tarikh akses 7/7/2001.
- Simpson, Alan & Robinson, Celeste. (1999). *Mastering Access 2000* . Sybex Inc. California.
- Microsoft Press.(2001). *Easy Reference Guide Microsoft Access 2000*. New Horizon Publishing. Selangor



Hartman, Patricia. (2000). *Teach Yourself On The Spot Answer Visual Basic 6.0*. IDG Books Worldwide Inc. California.

Schneider, David I. (1999). *An Introduction to Programming Using Visual Basic 6.0*. Prentice Hall Inc. New Jersey.

Cornell, Gary. (1998). *Visual Basic 6.0 from the Ground Up*. Osborne/McGraw Hill. California.

# DIAGRAM ANAK UNTUK SETIAP PROSES DALAM DIAGRAM '0'

Diagram 1:

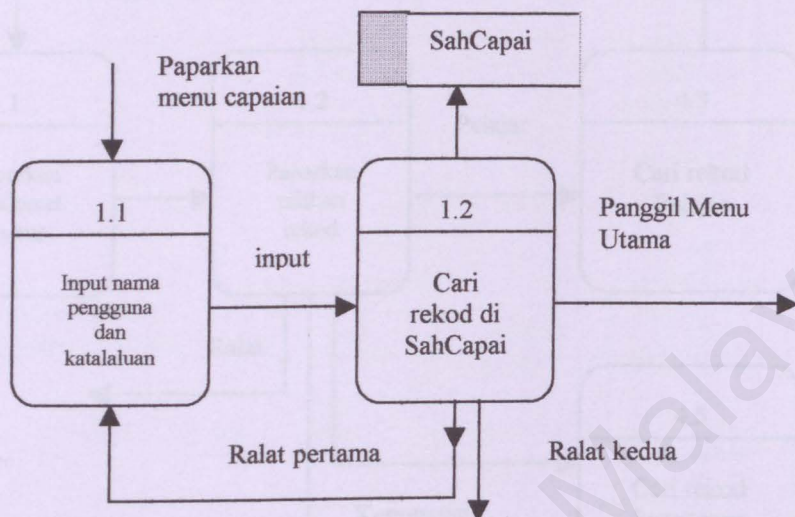


Diagram 3:

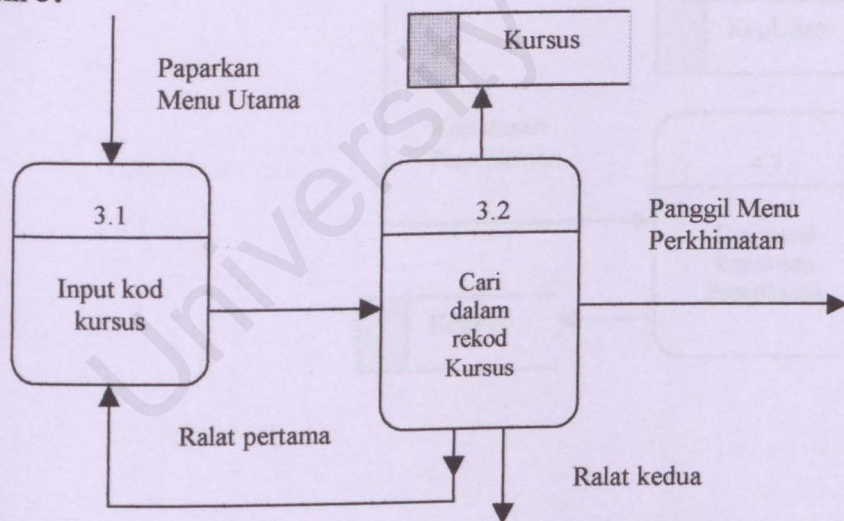




Diagram 3:

Diagram 4:

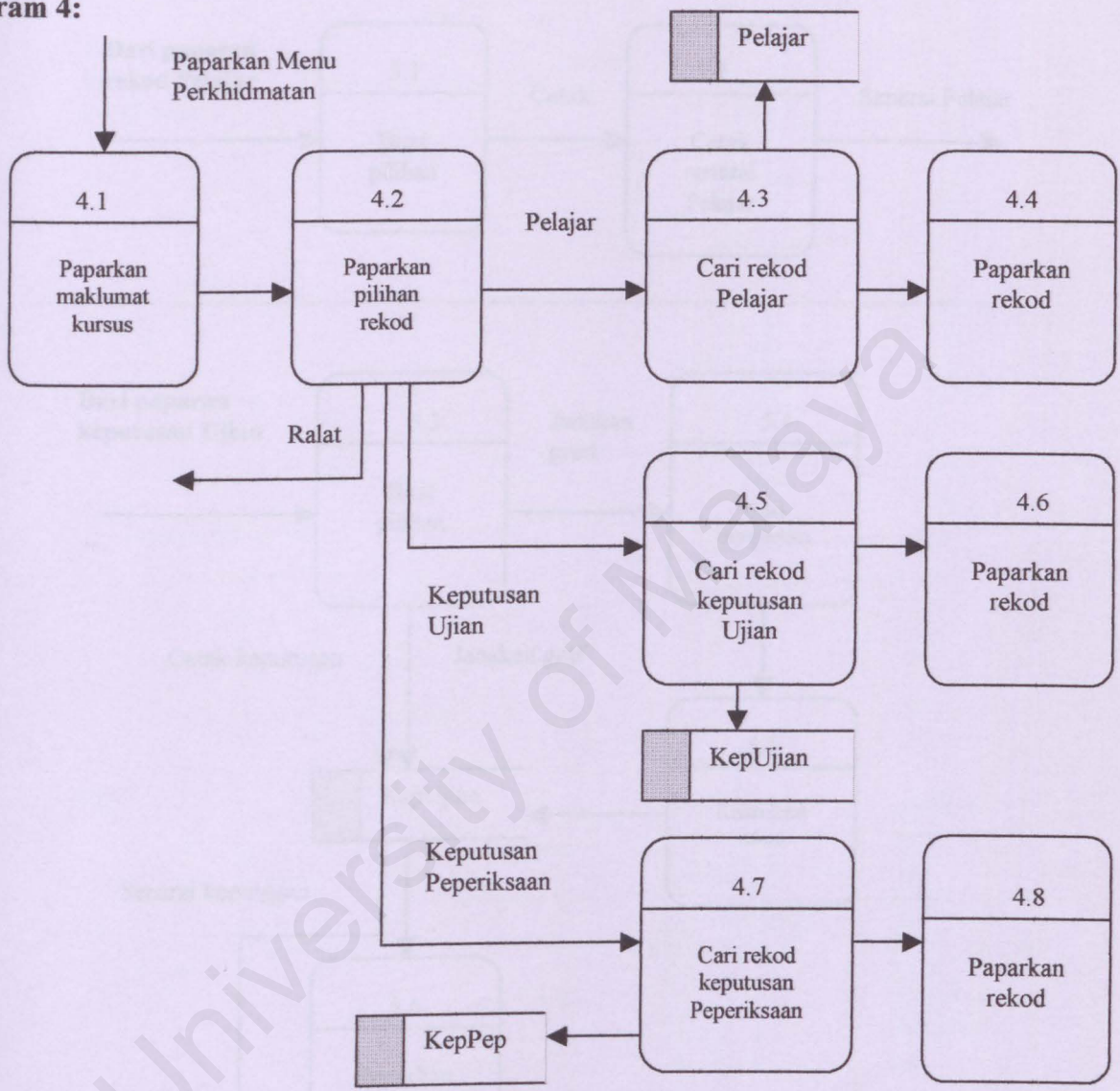
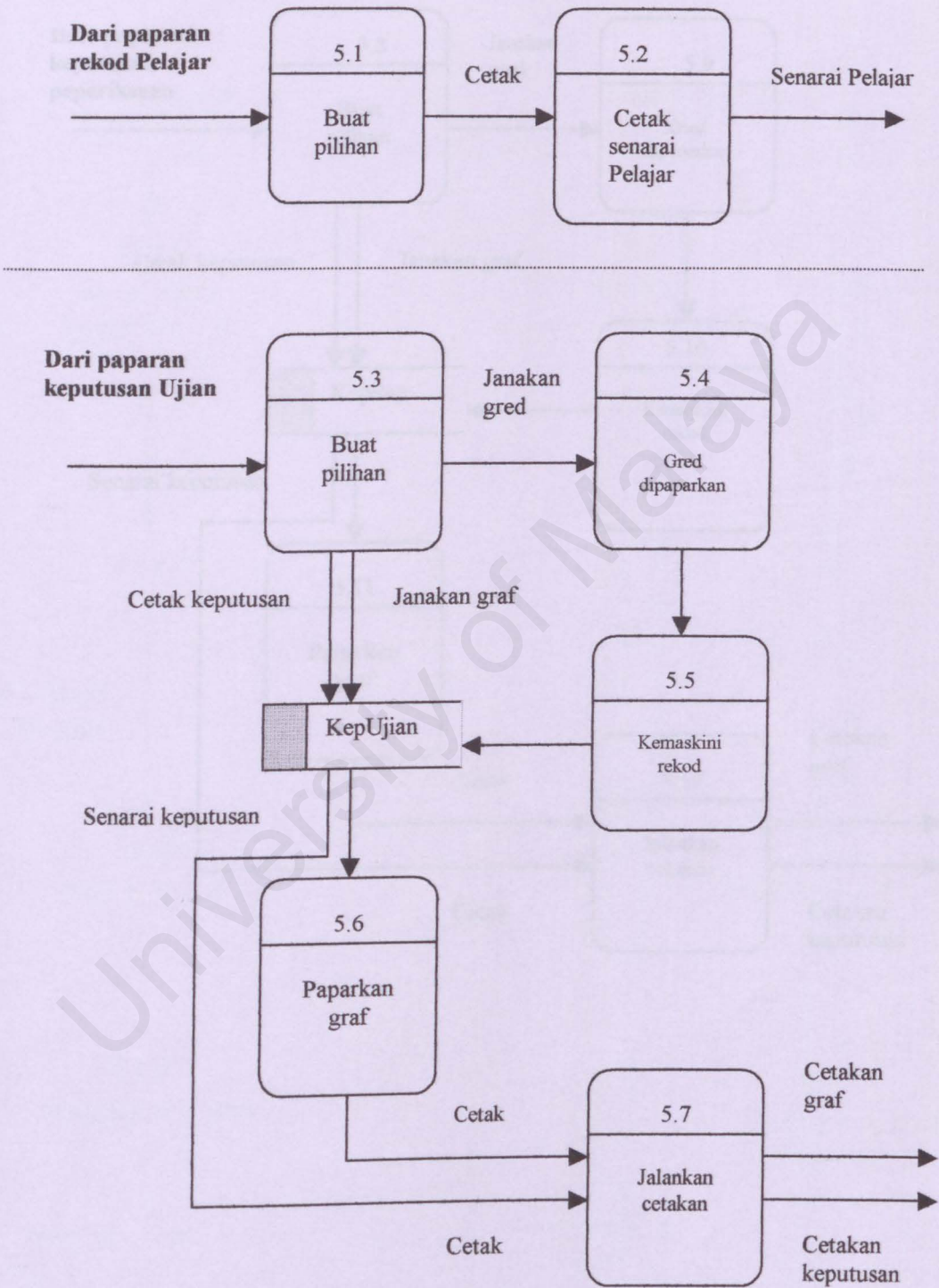
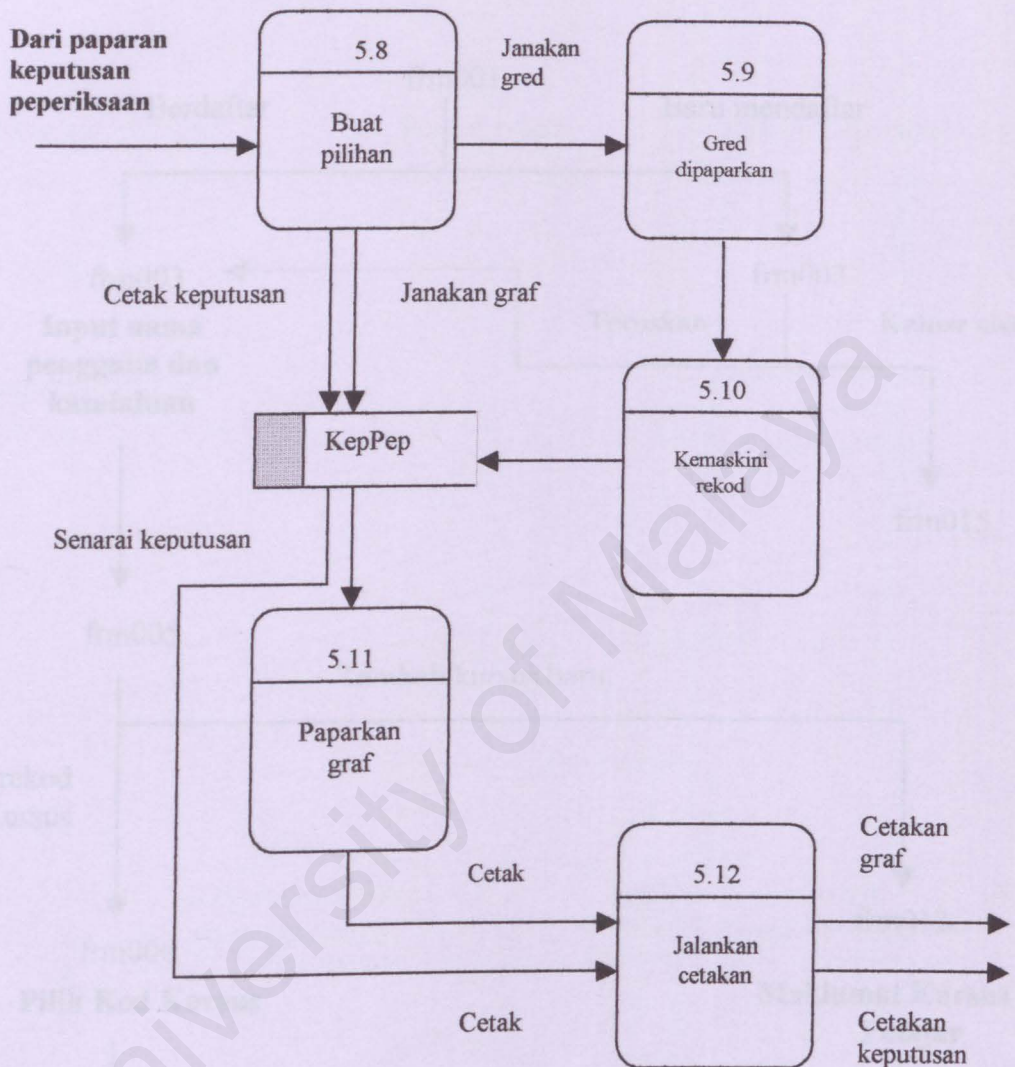


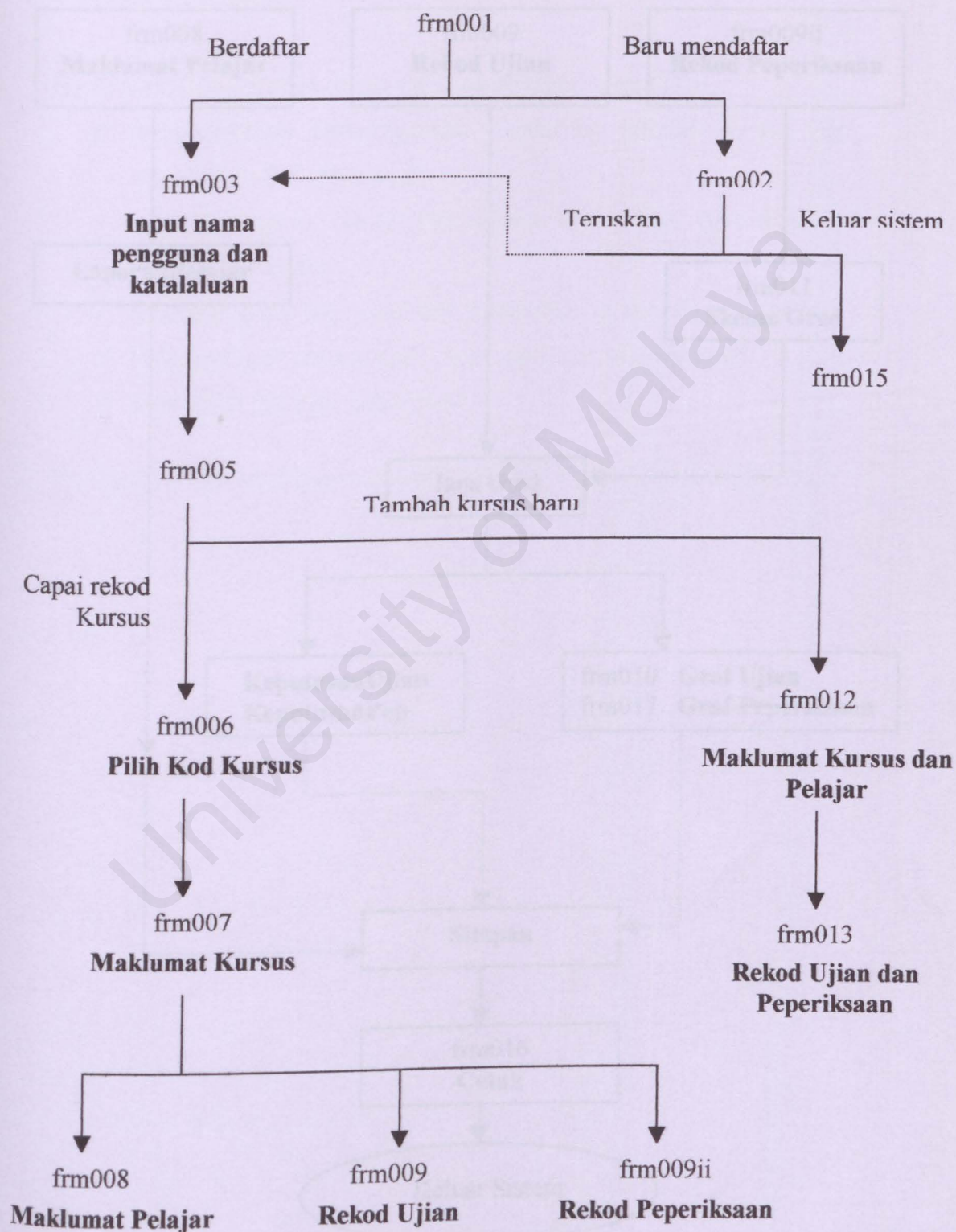
Diagram 5:





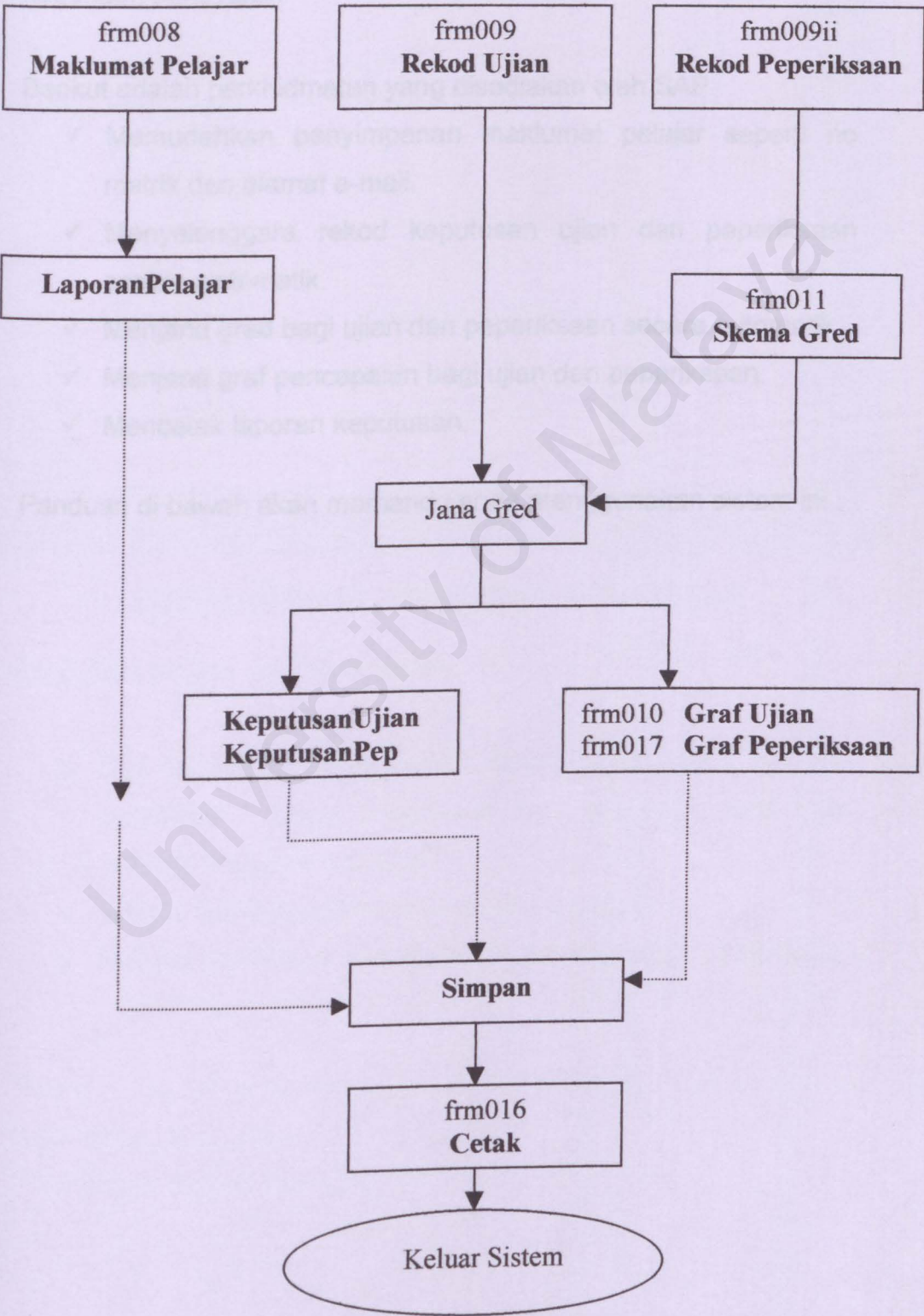


## PERGERAKAN BORANG DALAM SAP





BAGI SETIAP REKOD PELAJAR, REKOD UJIAN DAN REKOD PEPERIKSAAN:



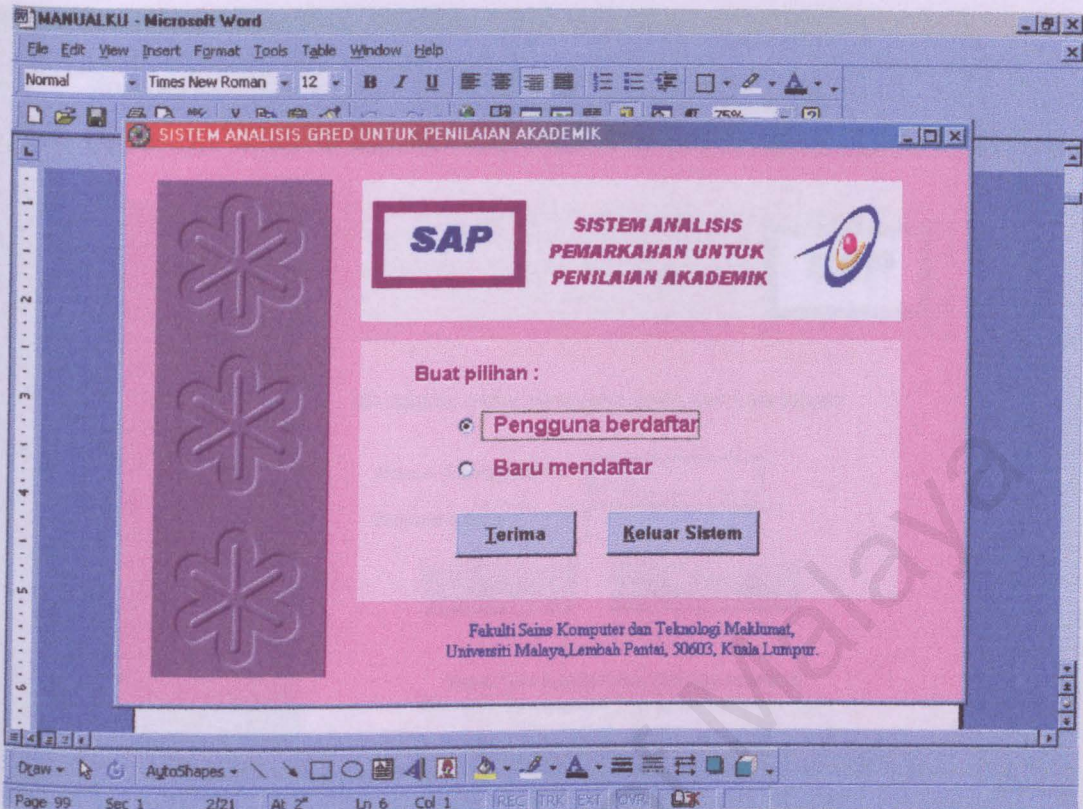
**SAP** atau Sistem Analisis Pemarkahan untuk Penilaian Akademik adalah suatu sistem sokongan yang dibangunkan khas untuk kegunaan pensyarah.

Berikut adalah perkhidmatan yang disediakan oleh SAP:

- ✓ Memudahkan penyimpanan maklumat pelajar seperti no matrik dan alamat e-mail.
- ✓ Menyelenggara rekod keputusan ujian dan peperiksaan secara sistematik.
- ✓ Menjana gred bagi ujian dan peperiksaan secara automatik.
- ✓ Menjana graf pencapaian bagi ujian dan peperiksaan.
- ✓ Mencetak laporan keputusan.

Panduan di bawah akan memandu anda menggunakan sistem ini....



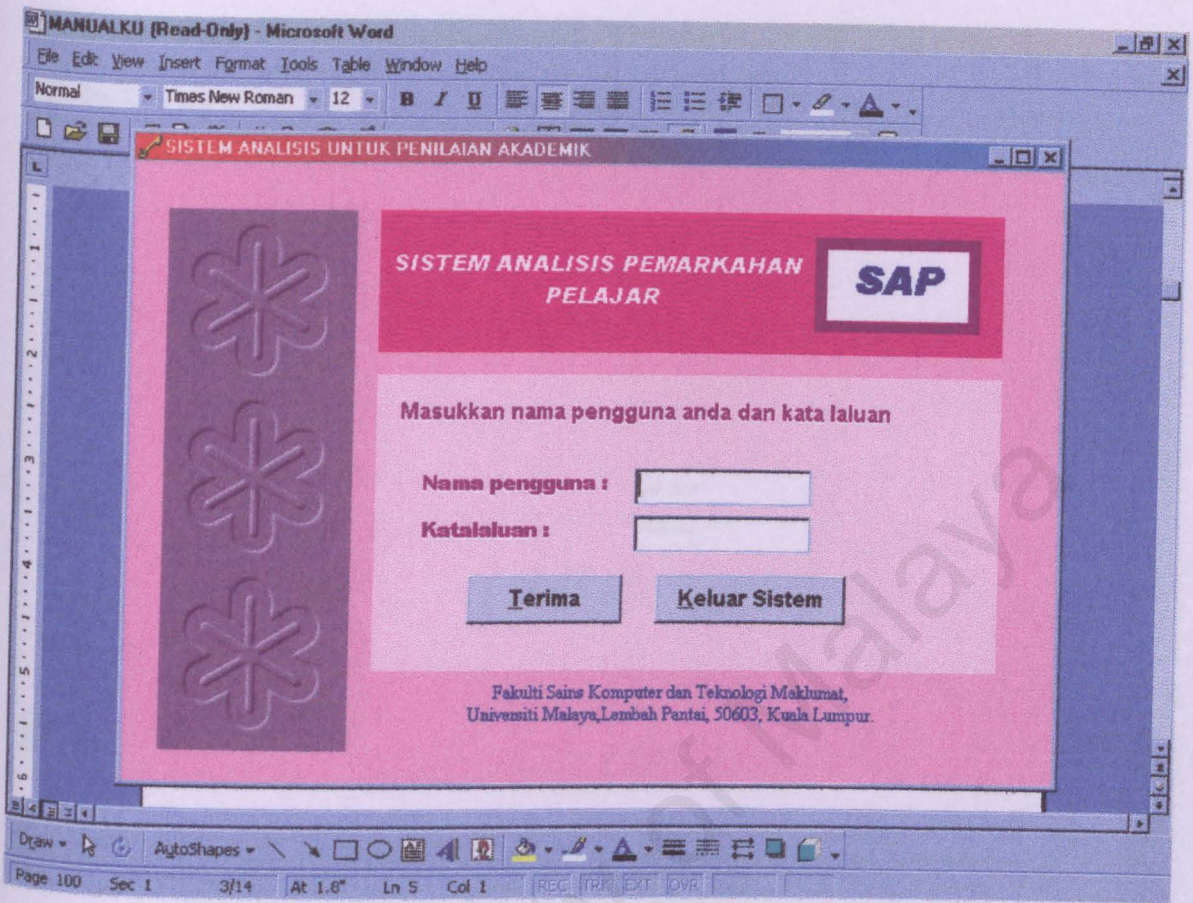


### SKRIN PERTAMA : PILIH JENIS KEMASUKAN

Ini adalah skrin pertama yang akan muncul.

- ❑ Oleh kerana anda baru hendak menggunakan sistem, sila pilih '**Baru mendaftar**'. Untuk penggunaan yang seterusnya, anda hanya perlu memilih 'Pengguna berdaftar'.
- ❑ Selepas membuat pilihan, tekan butang '**Terima**'.
- ❑ Pilih '**Keluar Sistem**' jika tidak berminat untuk meneruskan capaian.





## PENDAFTARAN PENGGUNA BUAT KALI PERTAMA

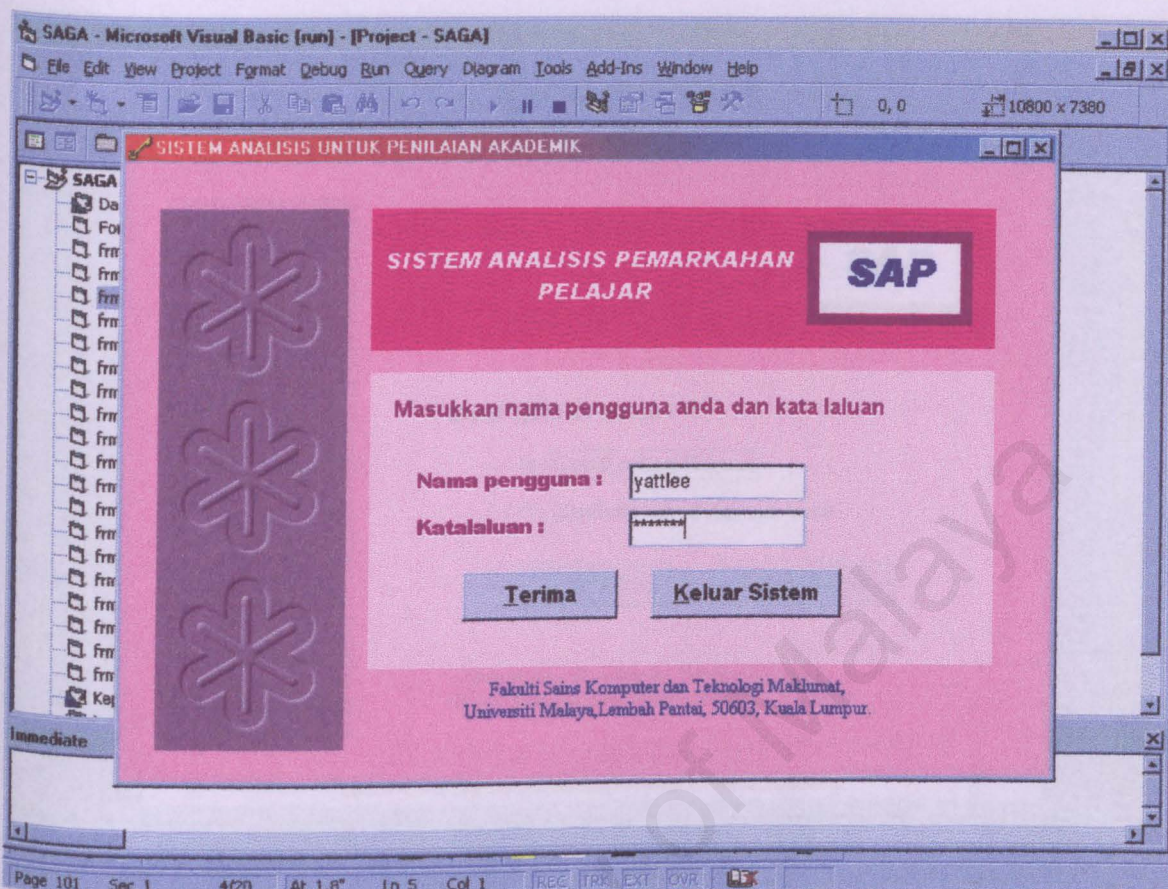
Sekiranya anda pertama kali mendaftar, skrin ini akan dipaparkan.

- ❑ Sila isikan maklumat yang dikehendaki:

Jika anda meletakkan *cursor* di kotak input, *tips* akan membantu anda memasukkan input yang boleh diterima. Bagi kedua-dua nama pengguna dan katalaluan, sebarang kombinasi aksara dan nombor diterima (tidak lebih 8 unit).

- ❑ Selesai mendaftar, tekan butang '**Daftar**' dan seterusnya tekan '**Teruskan**' jika ingin meneruskan capaian.
- ❑ Pilih '**Keluar Sistem**' jika ingin keluar dari sistem.



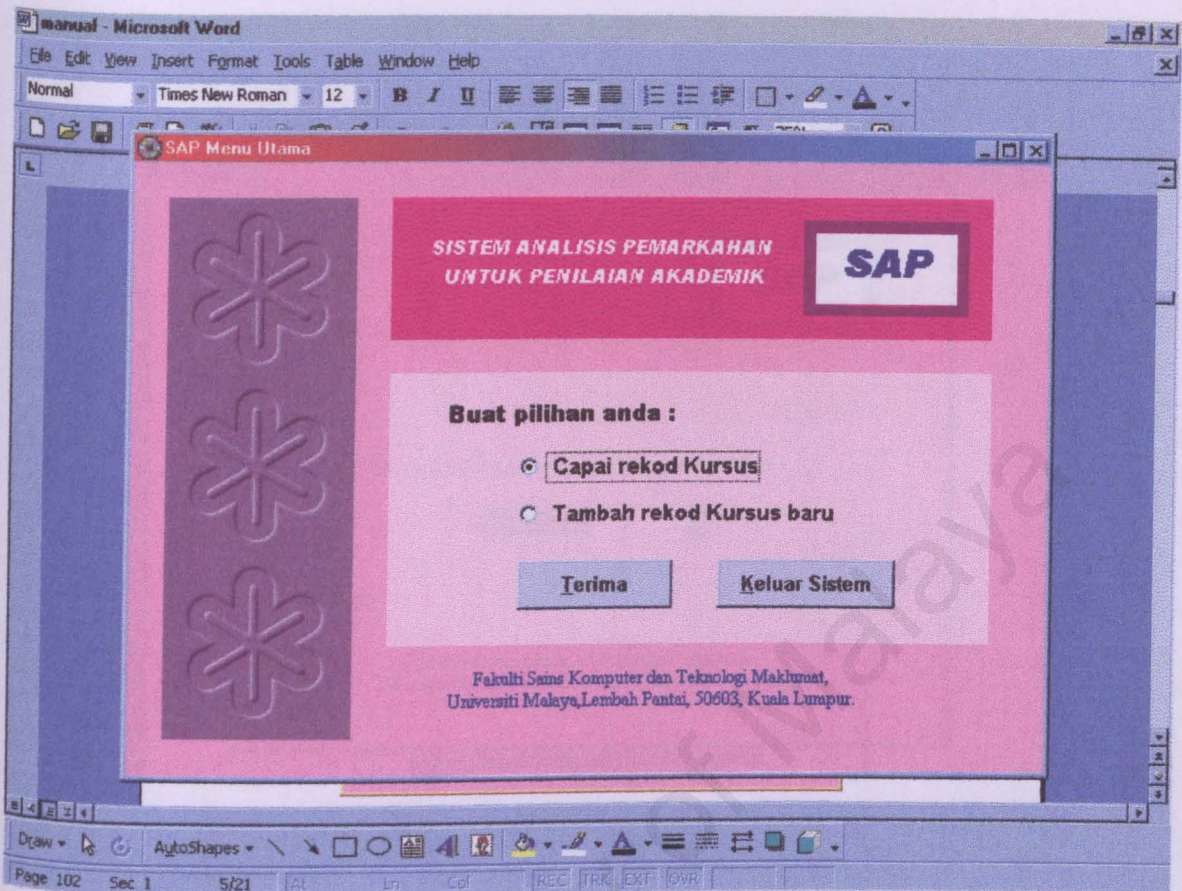


**SKRIN LOG-IN : MASUKKAN NAMA PENGGUNA DAN KATALALUAN**

Skrin ini dipaparkan sekiranya anda memilih untuk meneruskan capaian.

- ❑ Masukkan nama pengguna dan katalaluan anda yang telah disahkan dan tekan butang '**Terima**'.
- ❑ Jika input tidak sah, mesej ralat akan dipaparkan dan anda diminta menginput nama pengguna dan katalaluan sekali lagi.



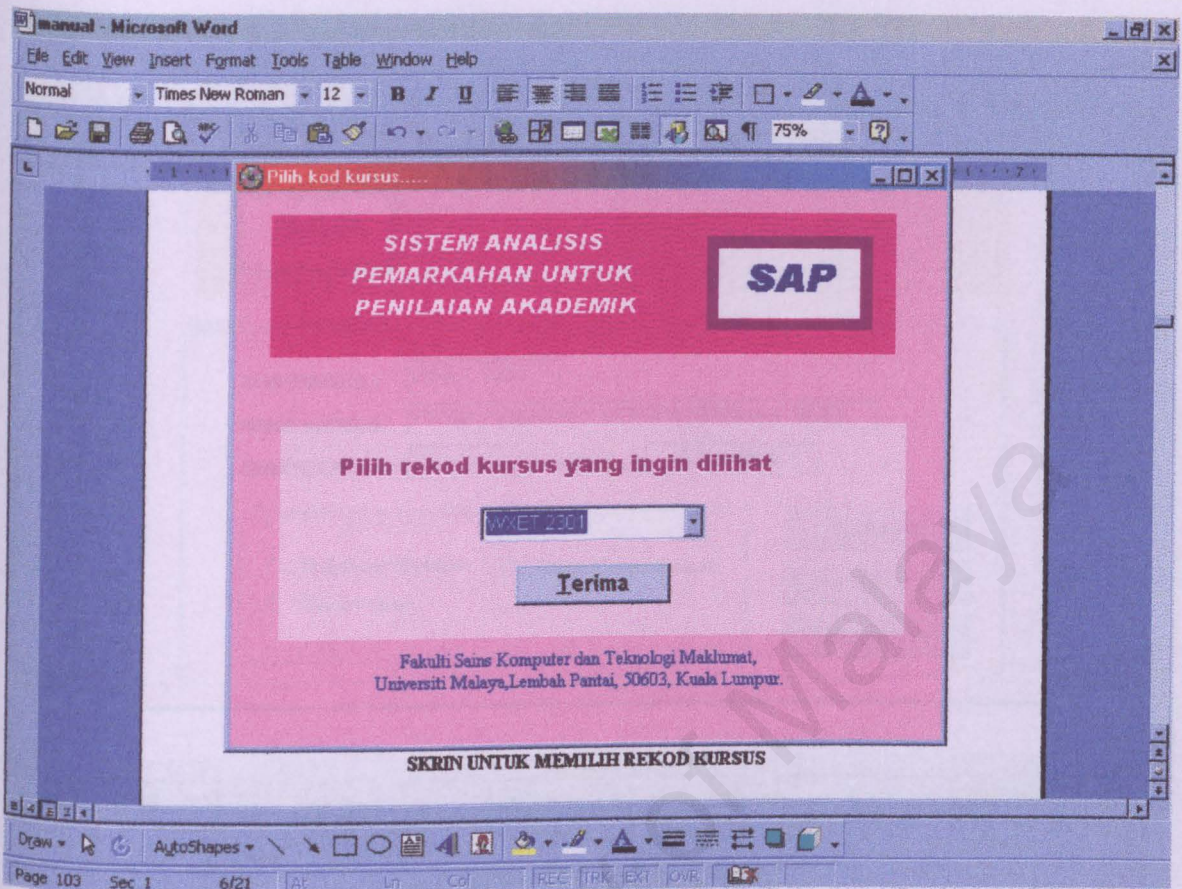


### SKRIN PILIH

Jika nama pengguna dan kata laluan telah disahkan, skrin pilihan ini dipaparkan.

- ❑ Sekiranya anda ingin melihat rekod yang sedia ada, pilih '**Capai rekod Kursus**' dan tekan '**Terima**'.
- ❑ Sekiranya anda ingin menambah rekod kursus yang baru, pilih '**Tambah rekod Kursus baru**' dan tekan '**Terima**'.



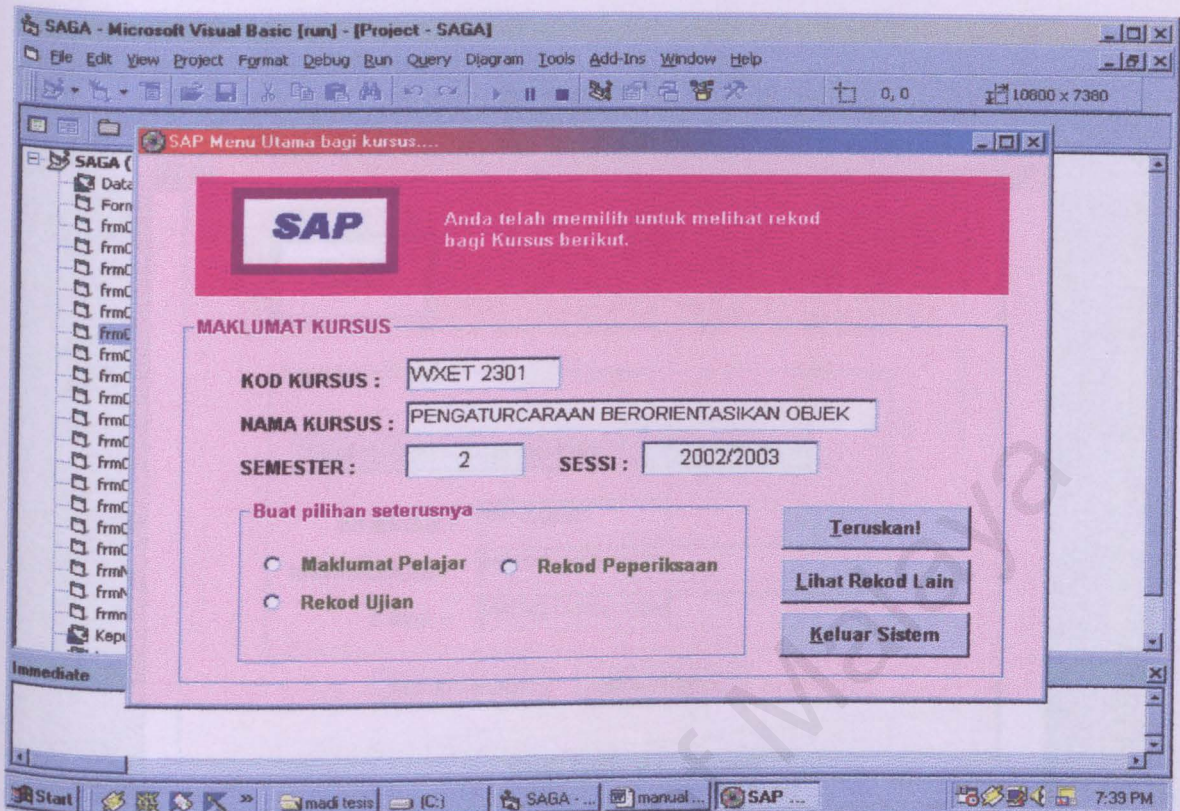


### SKRIN UNTUK MEMILIH REKOD KURSUS

Sekiranya anda memilih untuk melihat rekod sedia ada, skrin ini dipaparkan kepada anda.

Scroll ke bawah *combo box* yang berada di tengah skrin untuk memilih rekod kod kursus yang ingin dilihat dan tekan 'Terima'.



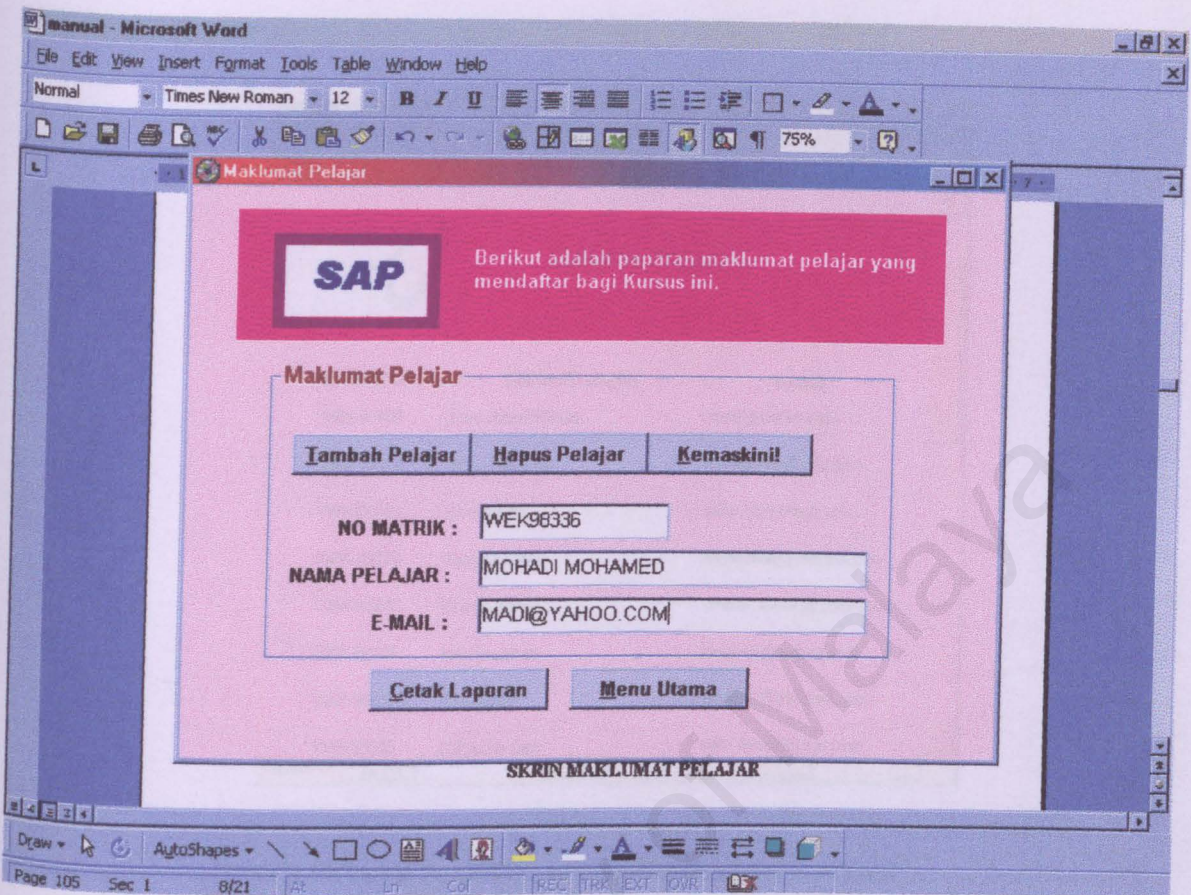


### MENU UTAMA ANDA!

Skrin ini dianggap sebagai menu utama anda.

- Maklumat berkenaan Kursus yang anda pilih tadi dipaparkan pada skrin.
  - Pilih '**Maklumat Pelajar**' jika ingin melihat maklumat pelajar yang mengambil kursus ini.
  - Pilih '**Rekod Ujian**' jika ingin melihat rekod keputusan ujian bagi kursus ini.
  - Pilih '**Rekod Peperiksaan**' jika ingin melihat rekod keputusan peperiksaan bagi kursus ini.
  - Selepas membuat pilihan, tekan '**Teruskan!**'.
  - Pilih '**Lihat rekod lain**' jika ingin melihat rekod kursus lain yang ada dalam simpanan.
  - Pilih '**Keluar Sistem**' untuk keluar dari sistem.





### SKRIN MAKLUMAT PELAJAR

Sekiranya anda memilih untuk melihat maklumat pelajar, skrin ini kan dipaparkan. *Scroll* papan kawalan data untuk membaca maklumat semua pelajar yang berdaftar untuk kursus berkenaan.

- ❑ Untuk menambah rekod pelajar, klik '**Tambah Pelajar**'. Ruangan input akan dikosongkan dan anda boleh mula menginput maklumat pelajar baru. Kemudian tekan '**Kemaskini!**' untuk menyimpan data yang baru ditambah itu.
- ❑ Untuk membuang rekod mana-mana pelajar, klik '**Hapus Pelajar**'. Data tersebut akan terus dihapuskan dan rekod berganjak ke rekod pelajar yang seterusnya. Kemudian tekan '**Kemaskini!**' untuk mengemaskini rekod.
- ❑ Pilih '**Cetak Laporan**' untuk mencetak salinan maklumat pelajar.
- ❑ Pilih '**Menu Utama**' untuk kembali ke menu utama tadi.



**Laporan Senarai Pelajar**

Zoom: 100%

**SENARAI PELAJAR**

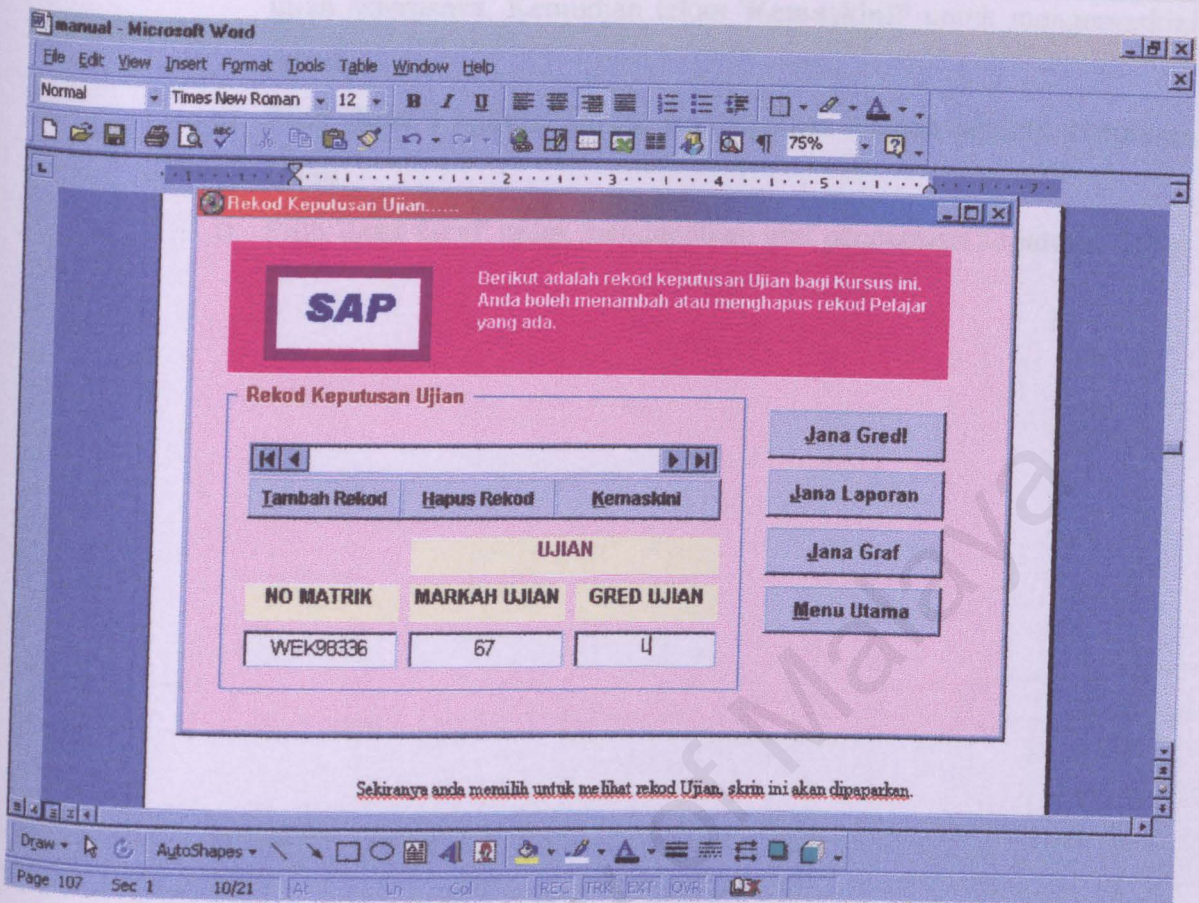
NO MATRIK	NAMA PELAJAR	E-MAIL
WEK 00103	Zuhir Ahmad Daud	zuhir@yahoo.com
WEK 00117	Wang Ser Wey	serwey80@tong.com
WEK 00123	Sharifah Ali	sherry@hotmail.com
WEK 00333	Kang Lee May	kayleemay@meihua.com
WEK 00808	Dinesh Kumar	dinesh_kumar@yahoo.co
WEK 98001	Mastura Musa	mastura@hotmail.com
WEK 98100	Tan Sherly	shelley_t@huahua.com
WEK 98122	Catherine Law	cate_law@yahoo.com

Pages: 1

### CONTOH LAPORAN SENARAI PELAJAR

- ❑ Untuk mencetak, klik ikon pencetak yang terdapat pada penghujung kiri atas skrin ini. Sekiranya anda ingin Untuk mencetak laporan ini, klik ikon pencetak pada penjuru kiri atas skrin ini.
- ❑ Untuk mengeksport laporan ini ke format lain dan direktori lain, klik ikon fail di sebelah ikon pencetak tadi.
- ❑ Untuk keluar dari skrin ini, klik sahaja tanda pangkah pada penjuru atas kanan skrin ini dan skrin maklumat pelajar tadi berada di hadapan anda.
- ❑ Klik '**Menu Utama**' untuk memilih untuk melihat rekod Ujian atau Peperiksaan pula.





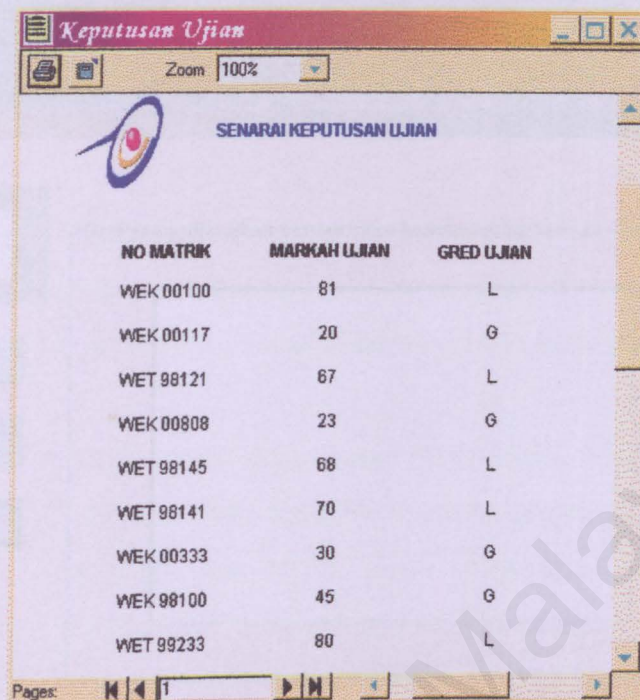
### SKRIN REKOD KEPUTUSAN UJIAN

Sekiranya anda memilih untuk melihat rekod Ujian, skrin ini akan dipaparkan. Scroll papan kawalan data untuk membaca keputusan Ujian bagi setiap pelajar.

- ❑ Untuk menambah rekod keputusan, klik '**Tambah Pelajar**'. Ruangan input akan dikosongkan dan anda boleh mula menginput no matrik dan markah Ujian sahaja.
- ❑ Tekan '**Jana Gred**' untuk menjanakan gred keputusan Ujian. Bila gred telah dipaparkan, tekan "**Kemaskini**" untuk menyimpan rekod yang sudah digredkan.
- ❑ Untuk membuang rekod mana-mana pelajar, klik '**Hapus Pelajar**'. Data tersebut akan terus dihapuskan dan rekod berganjak ke rekod



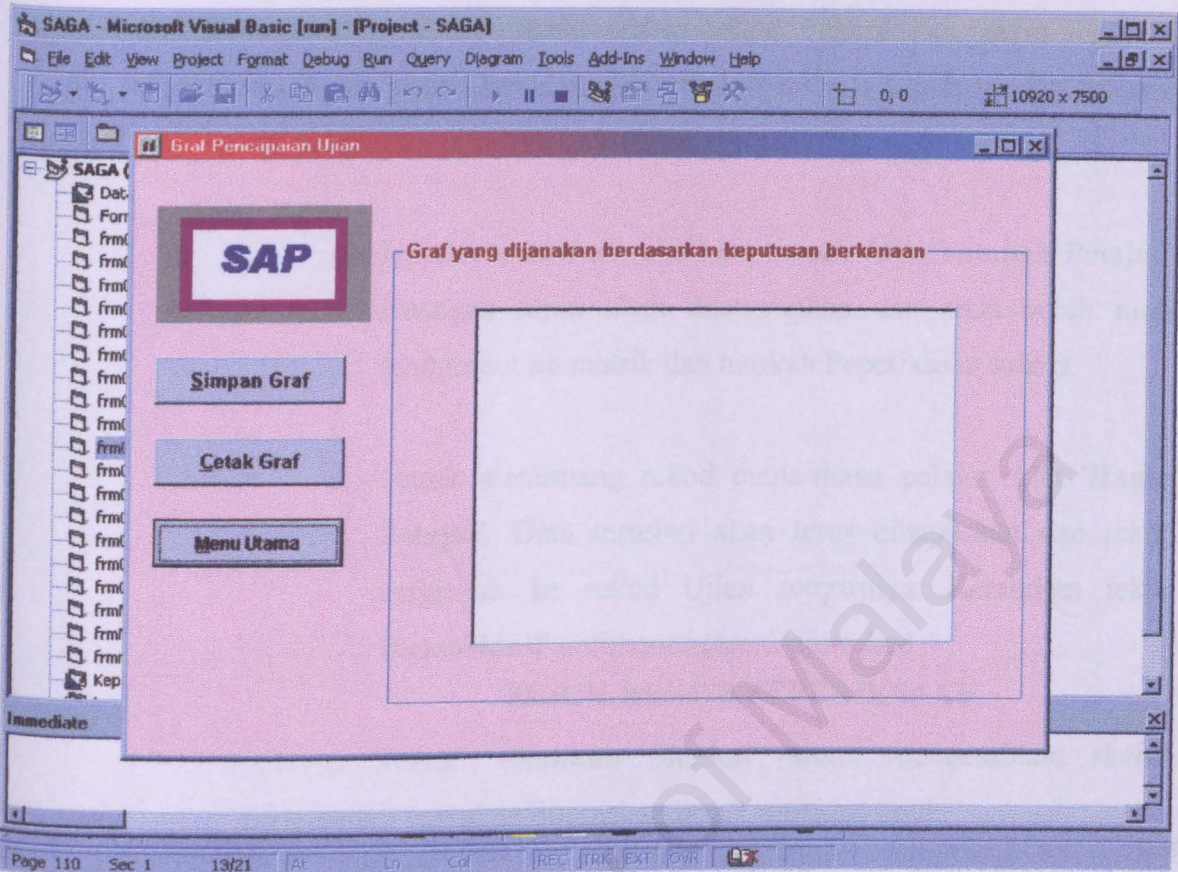




NO MATRIK	MARKAH UJIAN	GRED UJIAN
WEK 00100	81	L
WEK 00117	20	G
WET 98121	67	L
WEK 00808	23	G
WET 98145	68	L
WET 98141	70	L
WEK 00333	30	G
WEK 98100	45	G
WET 99233	80	L

### CONTOH LAPORAN KEPUTUSAN UJIAN YANG DIJANAKAN

- Untuk mencetak laporan ini, klik ikon pencetak
- Untuk mengeksport laporan ini ke format lain dan direktori lain, klik ikon fail
- Untuk keluar dari skrin ini, klik sahaja tanda pangkah pada penjuru atas kanan skrin ini dan skrin keputusan Ujian tadi berada di hadapan anda.



### SKRIN GRAF PENCAPAIAN GRAF UJIAN

Sekiranya anda memilih untuk menjana graf Ujian tadi, skrin yang memaparkan graf ini akan dipaparkan.

- Tekan '**Simpan Graf**' untuk menyimpan graf ini ke dalam direktori yang telah dispesifikkan.
- Tekan '**Cetak Graf**' untuk mencetak graf menggunakan pencetak yang telah dihubungkan secara automatik kepada komputer.
- Tekan '**Menu Utama**' untuk melihat rekod Peperiksaan pula.

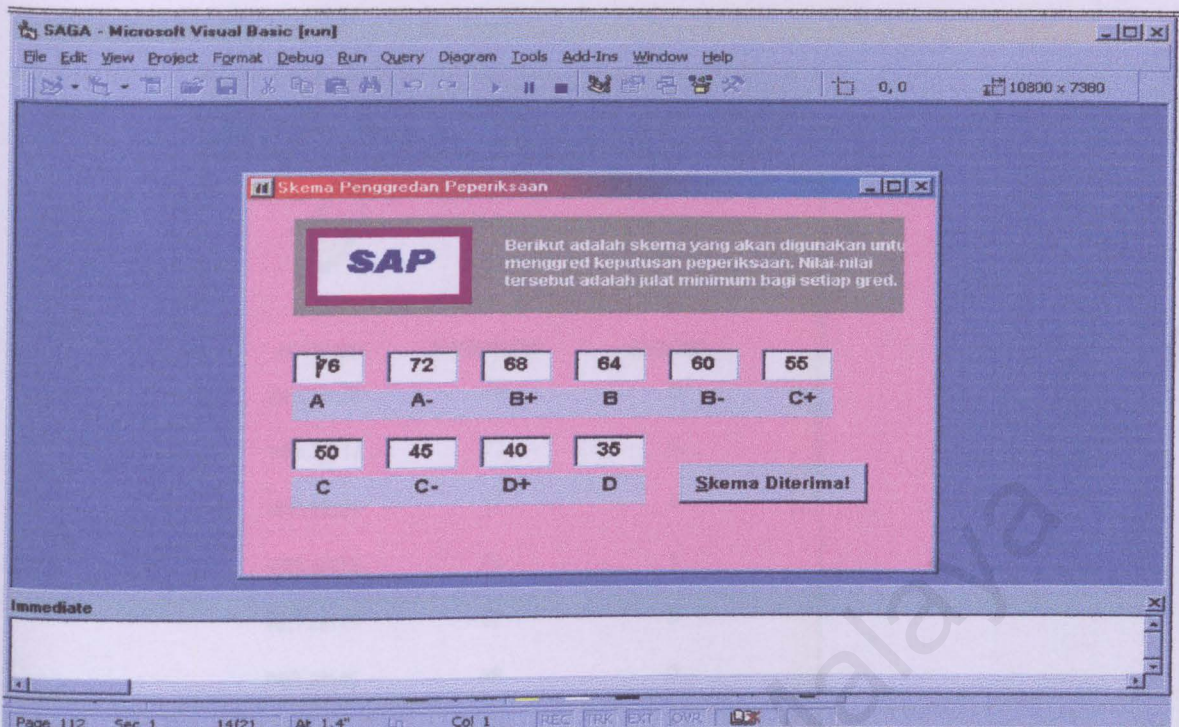


Sekiranya anda memilih untuk melihat rekod Peperiksaan, skrin ini akan dipaparkan. *Scroll* papan kawalan data untuk membaca keputusan Peperiksaan bagi setiap pelajar.

- ❑ Untuk menambah rekod keputusan, klik '**Tambah Pelajar**'. Ruangan input akan dikosongkan dan anda boleh mula menginput no matrik dan markah Peperiksaan sahaja.
- ❑ Untuk membuang rekod mana-mana pelajar, klik '**Hapus Pelajar**'. Data tersebut akan terus dihapuskan dan rekod berganjak ke rekod Ujian seterusnya. Kemudian tekan '**Kemaskini**!' untuk mengemaskini rekod

#### SKRIN REKOD PEPERIKSAAN

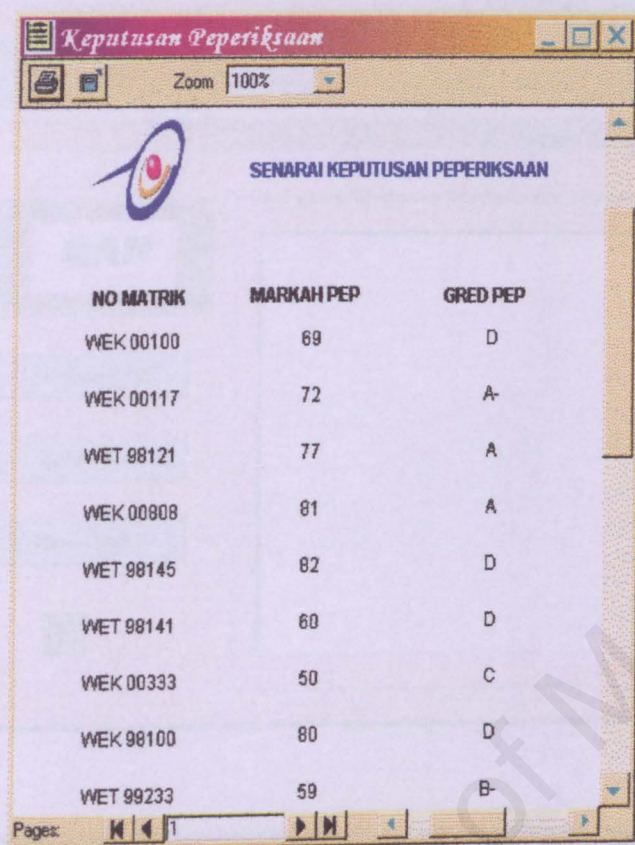
- ❑ Tekan '**Sahkan Skema**!' untuk mengesahkan skema penggredan yng akan digunakan untuk menjana gred.



## SKRIN SKEMA PENGGREDAN PEPERIKSAAN

- ❑ Tekan '**Skema Diterima**' untuk mengesahkan skema yang telah diiputkan.
- ❑ Kemudian, tekan '**Jana Gred**' untuk menjanakan gred keputusan Peperiksaan. Bila gred telah dipaparkan, tekan "**Kemaskini**" untuk menyimpan rekod yang sudah digredkan.
- ❑ Pilih '**Jana Laporan**' untuk mendapatkan salinan laporan keputusan Peperiksaan atau '**Jana Graf**' untuk mendapatkan graf pencapaian keputusan Peperiksaan.



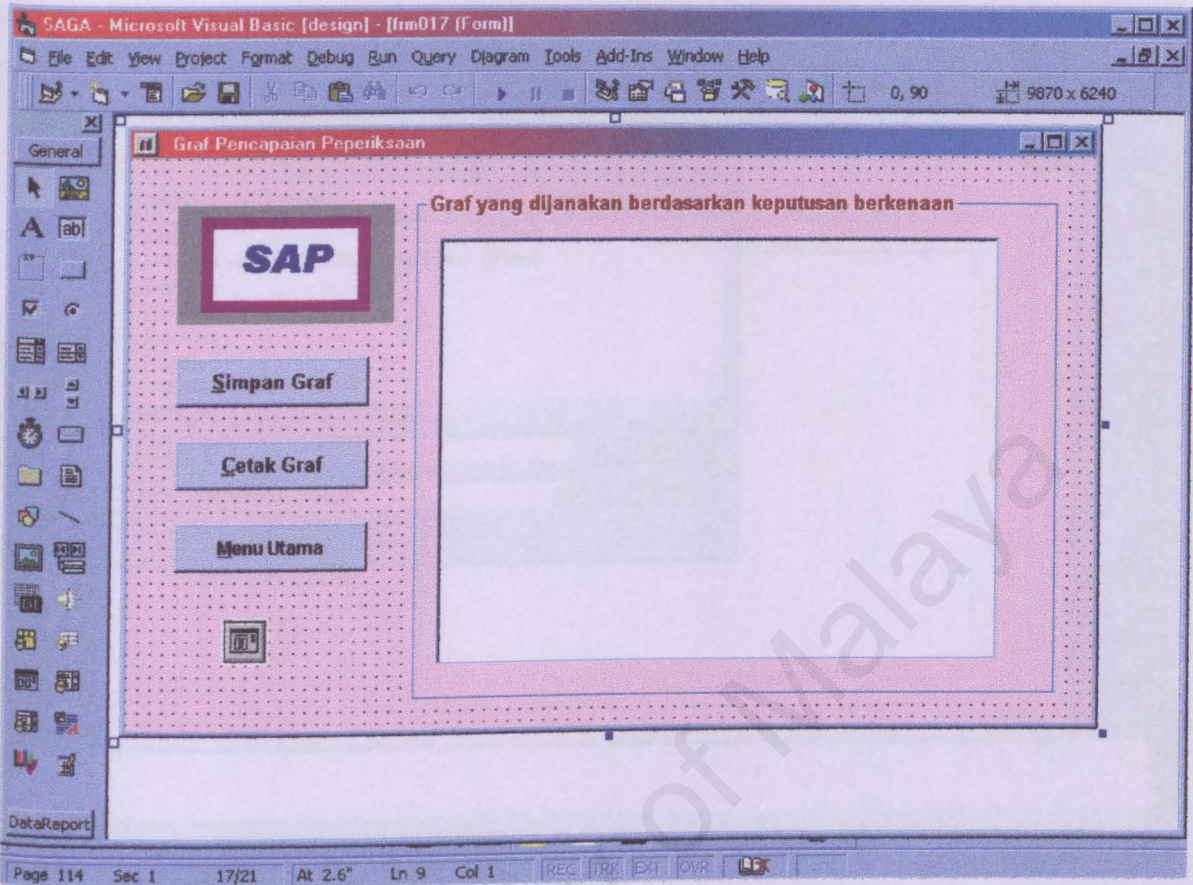


NO MATRIK	MARKAH PEP	GRED PEP
WEK 00100	69	D
WEK 00117	72	A-
WET 98121	77	A
WEK 00808	81	A
WET 98145	82	D
WET 98141	60	D
WEK 00333	50	C
WEK 98100	80	D
WET 99233	59	B-

### CONTOH LAPORAN KEPUTUSAN PEPERIKSAAN

- ✓ Untuk mencetak laporan ini, klik ikon pencetak
- ✓ Untuk mengeksport laporan ini ke format lain dan direktori lain, klik ikon fail
- ✓ Untuk keluar dari skrin ini, klik sahaja tanda pangkah pada penjuru atas kanan skrin ini dan skrin keputusan Ujian tadi berada di hadapan anda.

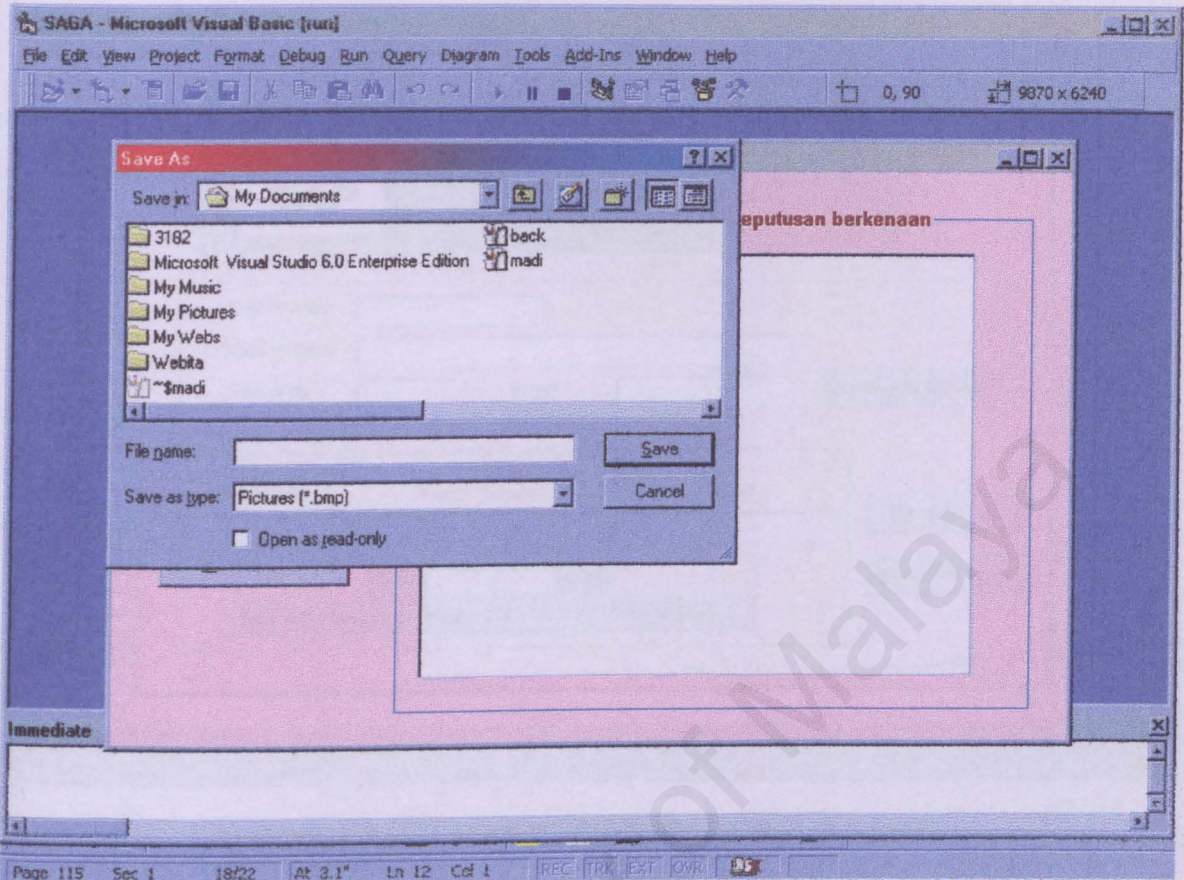




### SKRIN GRAF PENCAPAIAN PEPERIKSAAN

- Tekan '**Simpan Graf**' untuk menyimpan graf ini ke dalam direktori yang telah dispesifikan.
- Tekan '**Cetak Graf**' untuk mencetak graf menggunakan pencetak yang telah dihubungkan secara automatik kepada komputer.
- Tekan '**Menu Utama**' untuk melihat rekod Peperiksaan pula.

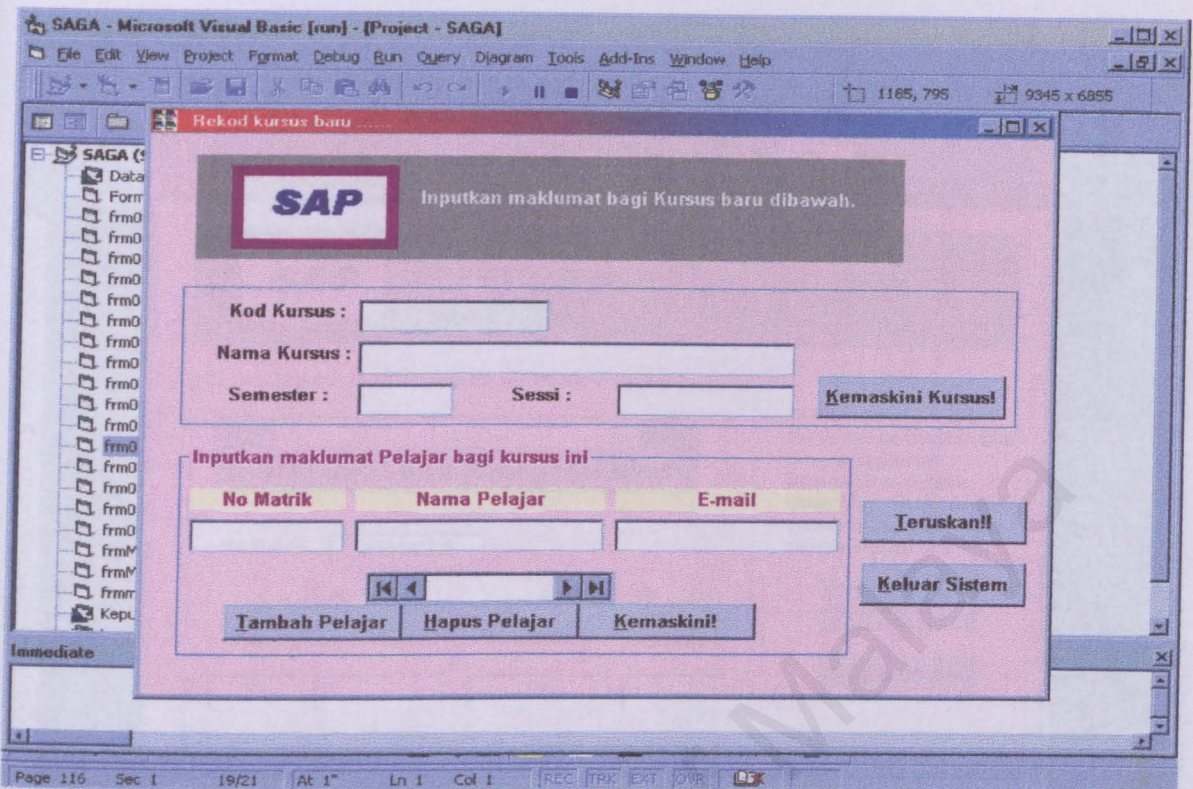




### SKRIN 'SAVE AS' UNTUK MENYIMPAN GRAF UJIAN & PEPERIKSAAN

- ✓ Tuliskan nama fail dan tekan 'Save' untuk menyimpan graf dalam direktori.





### SKRIN BAGI MEREKODKAN KURSUS BARU

Skrin ini akan dipaparkan di mana pengguna akan mula merekod bagi kursus baru yang ingin diwujudkan. Ia dibahagikan kepada dua bahagian.

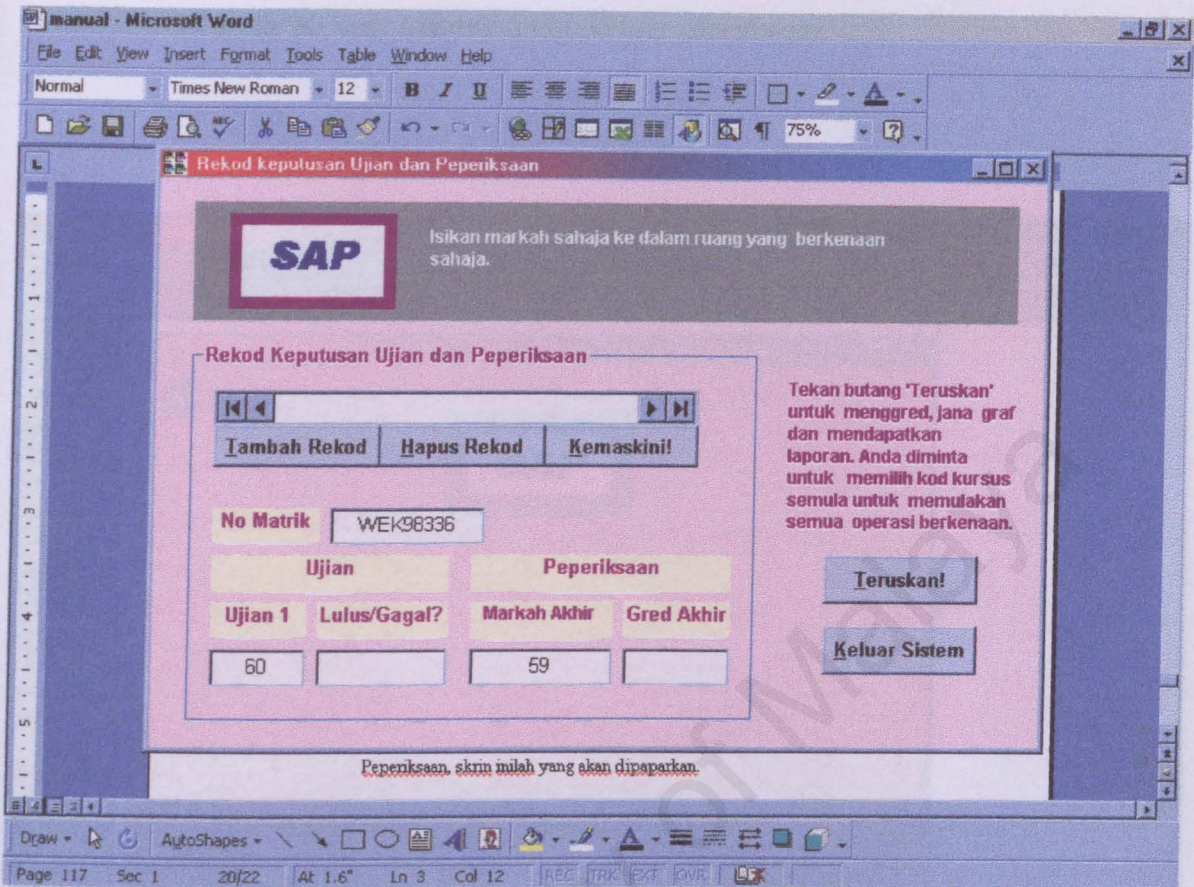
- Pada bahagian pertama, pengguna diminta mengisikan:

- Kod Kursus
- Semester
- Nama Kursus
- Sessi

Kemudian, tekan butang '**Kemaskini Kursus!**'.

- Pengguna seterusnya boleh menginput maklumat pelajar pula di bahagian kedua skrin. Setiap kali menambah atau menghapus rekod pelajar, pengguna mesti menekan butang '**Kemaskini!**' untuk menyimpan rekod.
- Tekan butang '**Teruskan!**' untuk menginput markah Ujian dan Peperiksaan pula atau '**Keluar Sistem**' untuk keluar dari sistem.





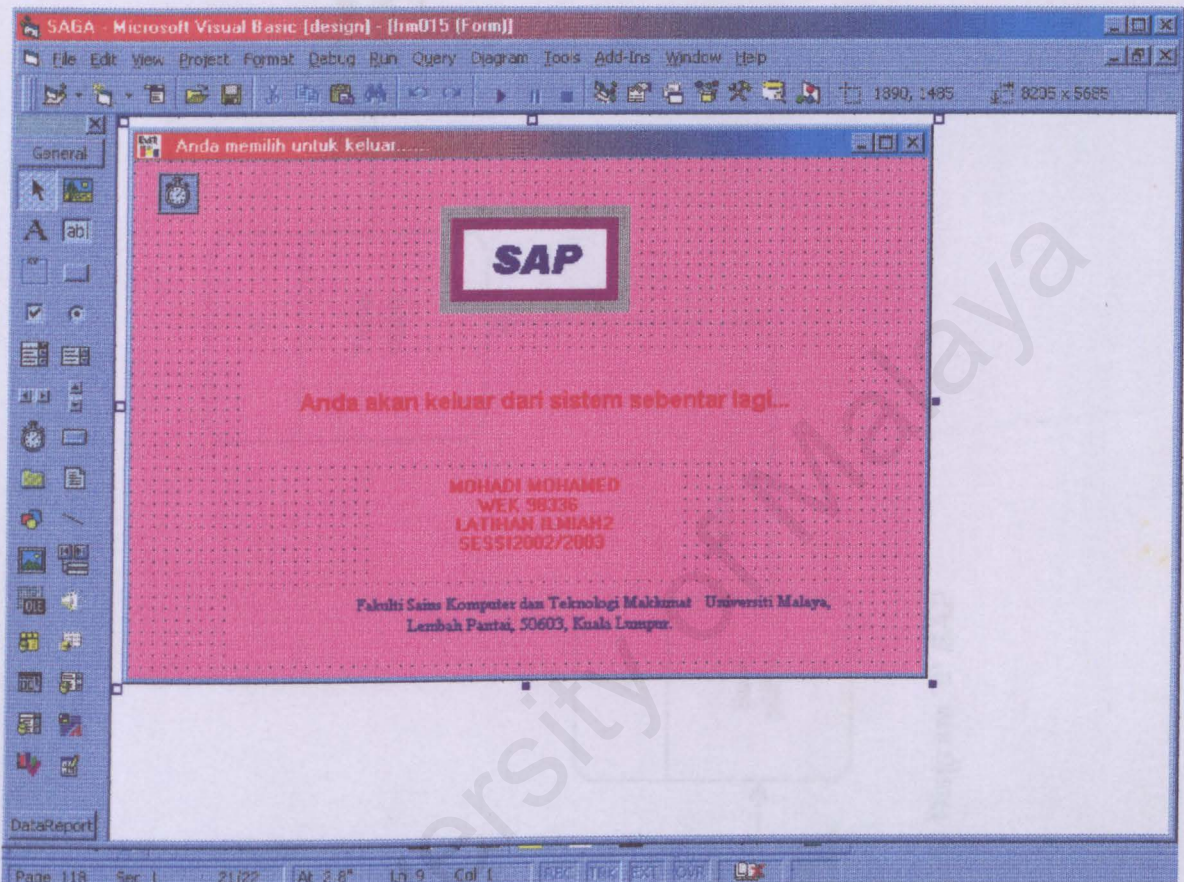
## SKRIN INPUT MAKLUMAT KEPUTUSAN BAGI REKOD BARU

Jika pengguna memilih untuk meneruskan input markah Ujian dan Peperiksaan, skrin inilah yang akan dipaparkan.

- Isikan no matrik, markah Ujian dan markah Peperiksaan sahaja.
- Tekan '**Teruskan**' jika ingin mengged, menjana graf dan mendapatkan laporan.
  - ✓ Dengan menekan butang ini, akan dibawa semula ke skrin memilih kod kursus dan anda diminta memilih semula kod kursus bagi kursus baru tadi yang ingin dilihat semula.
- Pilih '**Keluar Sistem**' untuk keluar dari sistem.



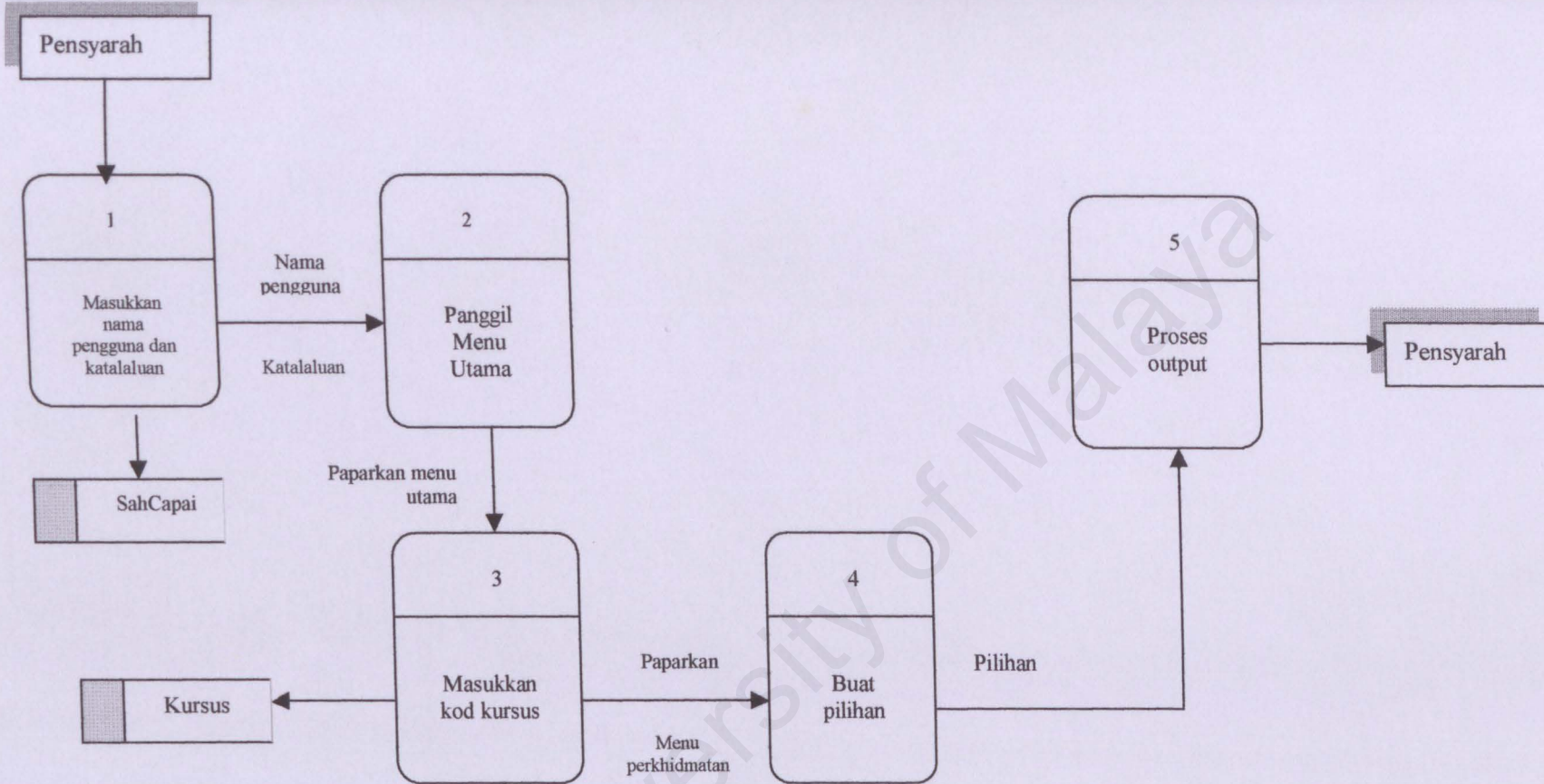
- ☐ Pilih '**Keluar Sistem**' untuk keluar dari sistem.



Setiap kali anda memilih untuk '**Keluar Sistem**', skrin inilah yang akan anda lihat. Dalam tempoh 12 saat, skrin ini akan ditutup dan menghasilkan sejenis kesan khas.

# SELAMAT MENGGUNAKAN SAP





Rajah 4.16 Diagram '0' SAP